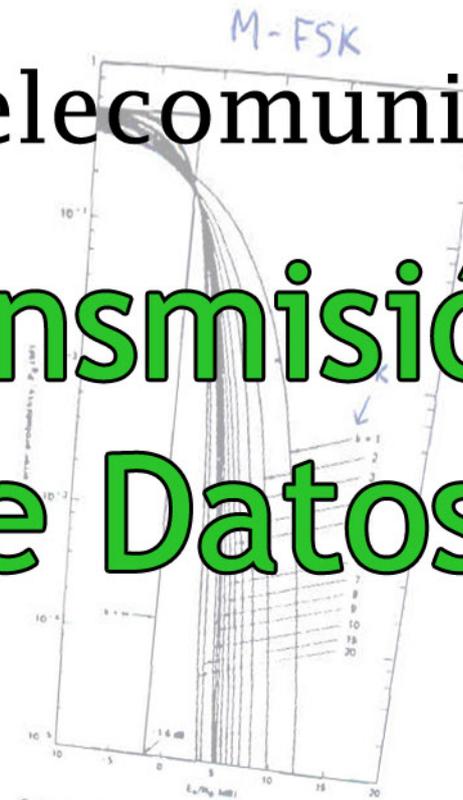
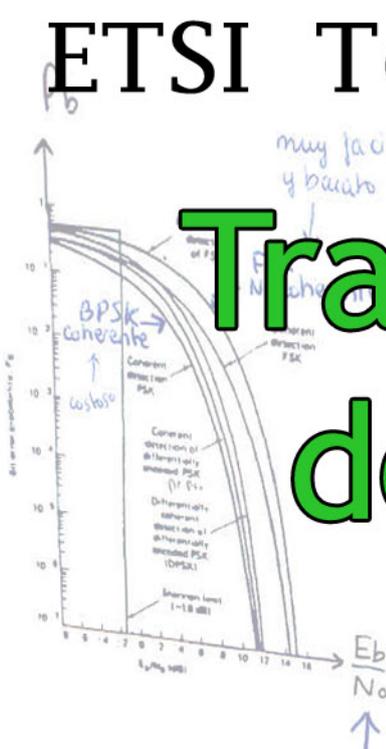
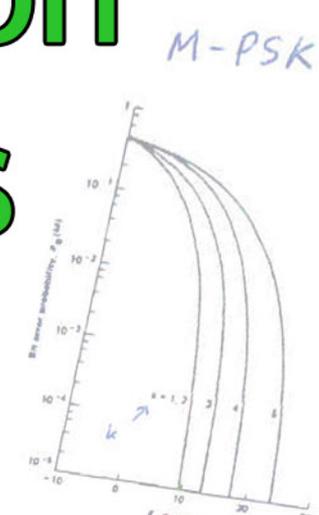


ETSI Telecomunicación

Transmisión de Datos

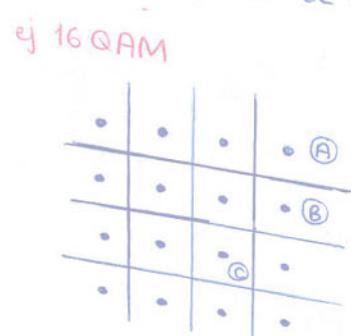


nota
 $k = n^{\circ}$ bits por simbolo
 $M = 2^k$



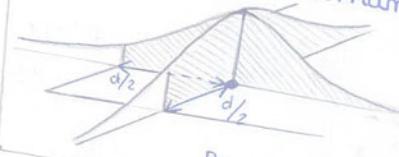
Curioso: al aumentar M baja la Pb i.e. requiero menos (S/N)
 Relación señal a ruido normalizada respecto a bit

Para comparar equitativamente
 $\frac{E_s}{N_0} = \log_2 M \frac{E_b}{N_0}$



Hay 3 tipos de regiones

Ⓐ $P_{eA} = 1 - P_{neA}$
 errores en A / no errores en A
 Para calcular P_{neA} tomamos como origen el punto

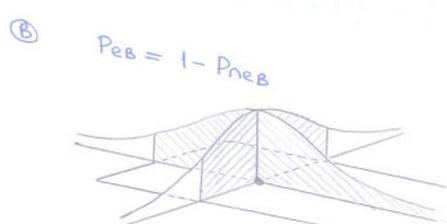


$$P_{neA} = (1 - Q(\frac{d}{2\sigma})) \cdot (1 - Q(\frac{d}{2\sigma}))$$

$$P_{eA} = 1 - P_{neA} = 1 - (1 - Q)(1 - Q) = 2Q - Q^2$$

denotamos $Q := Q(\frac{d}{2\sigma})$ por simplicidad

en logico es hacer $x < \frac{d}{2}$ OR $y < \frac{d}{2}$ MENOS la intersección ($x < \frac{d}{2}$ AND $y < \frac{d}{2}$) que habiamos contado doble



$$P_{neB} = (1 - Q) \cdot (1 - 2Q)$$

$$P_{eB} = 1 - (1 - Q) \cdot (1 - 2Q) = 3Q - 2Q^2$$

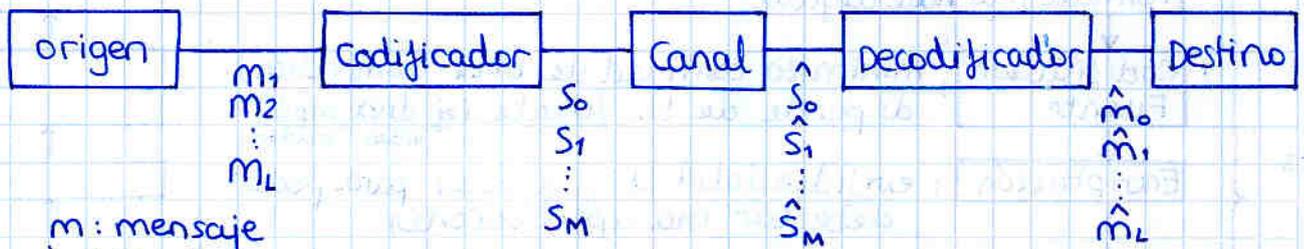
Transmisión de Datos

Apuntes de Pak (Francisco José Rodríguez Fortuño)
ETSI Telecomunicación. Universidad Politécnica de Valencia.
Primer cuatrimestre de 3^{er} curso
Curso 2005/2006

Fecha de última actualización: 21 Junio 2008

TEMA 1. Introducción a los sistemas de comunicación digital SCD

Modelo Simple de SCD



m : mensaje
ej: 1 caracter
256 posibilidades

s : señal
real susceptible
de ser transmitida

s es el representante en el canal de m ;
Pero la correspondencia no tiene porque ser 1 a 1 i.e. $L \neq M$

- un ' m ' puede ser varios ' s ': ej: $m = \text{caracter} = \text{byte}$; $s = \text{bit}$
- Varios ' m ' pueden ser un ' s ': ej: $s = \text{ASK de } M \text{ niveles}$; $m = \text{bit}$

Objetivos a cumplir

- máxima velocidad de transferencia
- mínimo ancho de banda
- mínima probabilidad de error
- mínima potencia en transmisor
- máxima utilización (nº usuarios, retardo)
- mínima complejidad y coste

los objetivos
'se pelean' entre
ellos.

Limitaciones

- Ancho de banda mínimo (Nyquist)
- Potencia mínima (Teorema Shannon)
- Leyes gubernamentales
- Limitaciones tecnológicas

Digital vs. Analógica

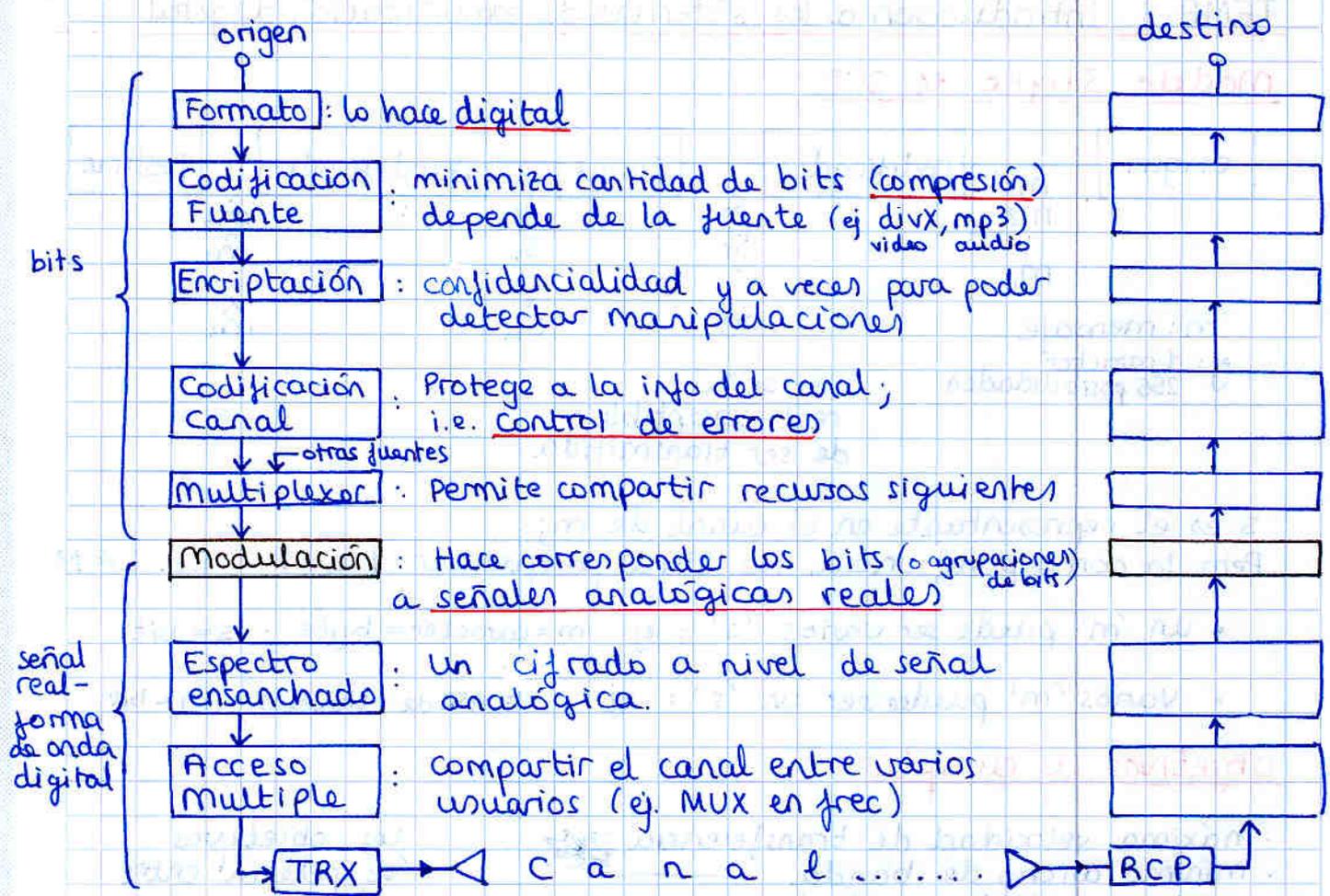
Ventajas:

- Interfaz Genérica
- Regeneración
- Efecto Umbral: si el ruido por debajo de un umbral \rightarrow comunicación perfecta.

Inconvenientes:

- Cuantificación \rightarrow pierde info
- Mayor ancho de banda

Modelo detallado



En recepción se siguen todos los pasos a la inversa. con una diferencia:

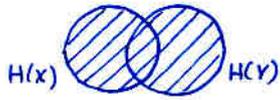
A ninguno de los niveles en recepción tenemos base de tiempos. Por ello se necesita **SINCRONIZACION**

Tema 2 - Teoría de la Información

• Información de un mensaje $I(x_i) = \log_2 \frac{1}{P(X=x_i)}$

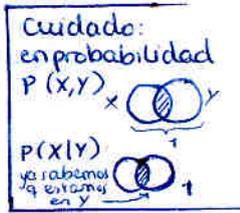
• Función entropía
 la suma ponderada de la $I(x_i)$ de sus mensajes
 $H(X) = \sum_{i=0}^{M-1} P(x=x_i) \cdot \log_2 \frac{1}{P(X=x_i)} \in [0, \log_2 M]$

• Entropía conjunta

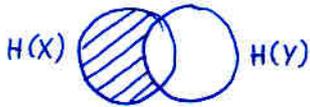


$$H(X, Y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i, Y=y_j)}$$

$\in [\max\{H(X), H(Y)\}, H(X) + H(Y)]$



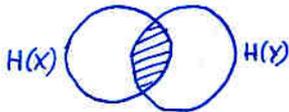
• Entropía condicional



$$H(X|Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i | Y=y_j)} \in [0, H(X)]$$

⚠ a la ponderación no le afecta la condicional

• Información mutua



$$\begin{aligned} I(X, Y) &= H(X) - H(X|Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log_2 \frac{P(X=x_i | Y=y_j)}{P(X=x_i)} \\ &= H(X) + H(Y) - H(X, Y) \\ &= I(Y, X) \end{aligned}$$

Para obtener $P(X=x_i, Y=y_j)$, $P(X=x_i | Y=y_j)$ usamos dos resultados (Normalmente en canal BSC conocemos $P(X=x_i)$ y $P(X \neq Y) = p$) i.e. $P(Y|X)$)

Teorema de Bayes

$\rightarrow P(X=x_i, Y=y_j) = P(Y=y_j | X=x_i) \cdot P(X=x_i)$ Para $P(X, Y)$

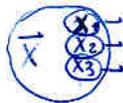
$\rightarrow P(X=x_i | Y=y_j) = \frac{P(X=x_i, Y=y_j)}{P(Y=y_j)}$ Para $P(X|Y)$

PERO necesitamos $P(Y=y_j)$
Teorema de prob. total

$$P(Y=y_j) = \sum_i P(X=x_i, Y=y_j)$$

es una suma!! no es un condicional

Vectores aleatorios discretos



si X_i son independientes
 $P(\bar{X}=\bar{x}) = \prod P(x_i=x_i)$
 $P(\bar{Y}=\bar{y} | \bar{X}=\bar{x}) = \prod P(y_i|x_i)$

$$I(\bar{X}, \bar{Y}) = \sum I(x_i=x_i | y_i=y_i)$$

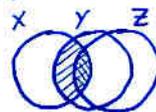
Teorema de Proceso de la Información

Cadena de Markov



Intuitivamente:
 Toda la información que X pueda dar a Z está en la que Y puede dar a Z

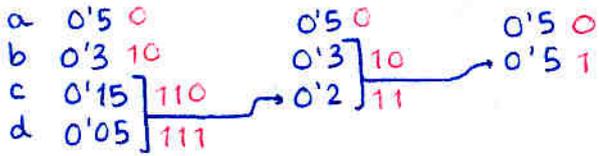
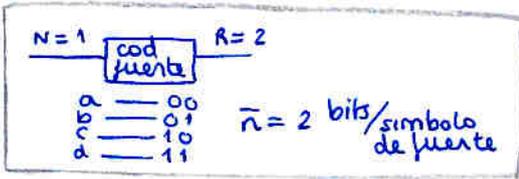
• $I(Z; X, Y) = I(Z; Y)$



• Además la información mutua entre 2 puntos viene limitada por la menor información mutua en un paso de la cadena entre los 2 puntos

Codificación de fuente

Algoritmo de Huffman



$$\bar{n} = 0.5 \cdot 1 + 0.3 \cdot 2 + 0.2 \cdot 3 = 1.7 \text{ bit/simbolo de fuente}$$

Se van agrupando de R en R (nº símbolos tras codif.) Excepto tal vez en el primer paso, donde hay que agrupar R^{k-1} tal que en el último paso asignemos los R símbolos. ($R^{k-1} \in [2, R]$)

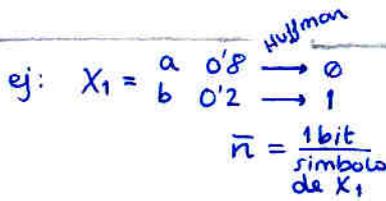
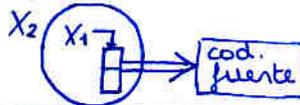
Truco: debe cumplirse $\text{Resto}\left(\frac{R^{k-1}}{R-1}\right) = \text{resto}\left(\frac{N}{R-1}\right)$

↓ tamaño del resto de grupos
↓ tamaño del 'primer grupo'
↓ nº de símbolos antes de codificación
↑ nº de símbolos tras codificación

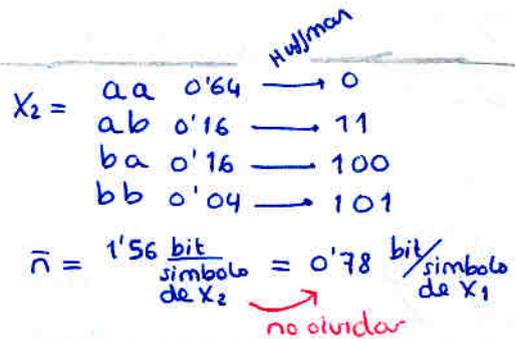
Teorema codificación de fuente de Shannon

$$\bar{n} \geq H(X)$$

Extensión de fuente:



hacemos extensión de fuente agrupando 2 símbolos



cuantos mas símbolos agrupemos más acercamos \bar{n} a su límite mínimo $H(X)$

Canales discretos

Capacidad:

$$C = \max(I(X, Y)) \quad \forall \text{ posibles fuentes } X$$

Para fuente vectorial:

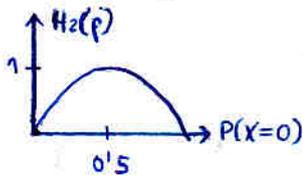
$$I(\bar{X}, \bar{Y}) = \sum_{i=1}^n I(X_i, Y_i) = n \cdot I(X_1, Y_1) \text{ si son iguales}$$

$$C_n = \max(I(\bar{X}, \bar{Y}))$$

$$C = \sup_{\forall n} \frac{1}{n} C_n$$

i.e. como fuente binaria, canal BSC

tomamos $\alpha = P(X=0)$ calculamos



$$I(X; Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log_2 \frac{P(X=x_i|Y=y_j)}{P(X=x_i)}$$

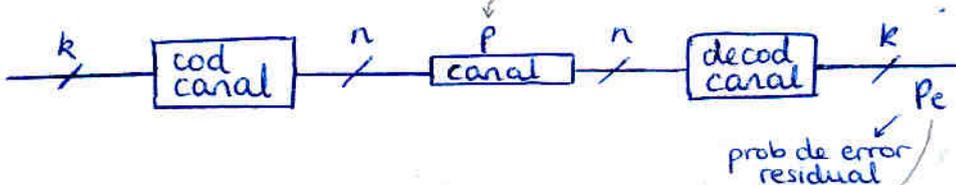
y lo maximizamos respecto a α

$$\frac{\partial I(X; Y)}{\partial \alpha} = 0 \Rightarrow \alpha = \frac{1}{2} \Rightarrow I(X; Y) = 1 - H_2(p)$$

⇒ canal BSC

$$C = 1 - H_2(p)$$

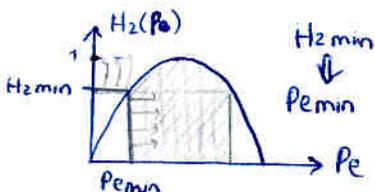
Ratio de codificación



$$R = \frac{k}{n}$$

↑ poca BW
↓ poca seguridad

Teorema de codificación de canal de Shannon



$$H_2(Pe) \geq 1 - \frac{C}{R}$$

$$R \leq \frac{C}{1 - H_2(Pe)}$$

interesante $R \leq C \Rightarrow H_{\text{min}} \leq 0 \Rightarrow P_{\text{min}} \geq 0$

P_e puede ser tan pequeño como queramos

C impone el límite superior de R para tx ideal $R = C - H_2(Pe) \geq 0$ sin errores
Actualmente se consigue $R \leq C$

TEMA 2. Teoría de la información

1. Información y su medida

1.1. Concepto de información

Comunicación: información de origen a destino
 ¿mejor forma de representarla? ¿y ASCII?
 No sería mejor .zip (reducir bits necesarios)
 ¿mejor forma de enviarla? ¿A que ritmo?

Gran libro: 1948 Shannon "A mathematical theory of communication"

1.2 Información de un mensaje



$$I(x_i) = \log_z \frac{1}{P(X=x_i)}$$

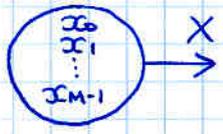
- menos probable → mas info
- si prob. se multiplican, la info se suma

Unidades:

según la base z:
 de ahora en adelante
 supondremos base 2 ←

z	Unidades
10	dits o Harleys
e	nats
2	bits o Shannons

1.3 Función entropía



mide lo caótica o impredecible que es una fuente X.

Se obtiene sumando la información de todos sus mensajes, ponderándolos según su probabilidad de aparición.

$$H(X) = \sum_{i=0}^{M-1} P(X=x_i) \cdot \log_2 \frac{1}{P(X=x_i)} = \sum P(X=x_i) \cdot I(x_i)$$

$$0 \leq H(X) \leq \log_2 M$$

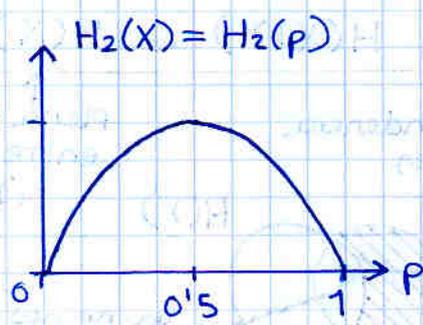
↑
 uno de los x_i tiene probabilidad 1, el resto 0 (muy predecible)

↑
 todos los mensajes equiprobables (impredecible)

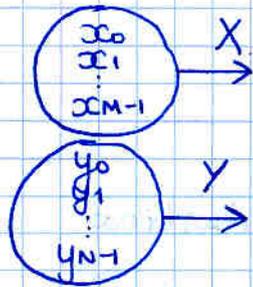
Caso particular: $z=2$ y fuente en binaria

$$P(X=x_0) = p$$

$$P(X=x_1) = 1-p$$



1.4 Entropía conjunta



$$I(x_i, y_j) = \log \frac{1}{P(X=x_i, Y=y_j)}$$

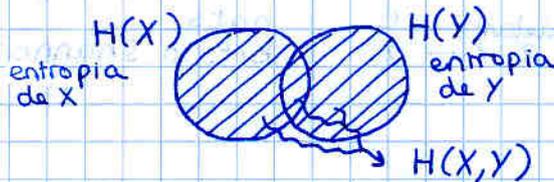
$$H(X, Y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i, Y=y_j)}$$

$$\max(H(X), H(Y)) \leq H(X, Y) \leq H(X) + H(Y)$$

↓
 todo lo que dice una fuente ya lo decía la otra:
 no hay incertidumbre adicional
 ej. leer 2 veces mismo libro de chistes

↓
 las fuentes no tienen nada que ver
 ej. leer libro de anatomía y libro de álgebra

Se puede representar como:



1.5 Entropía condicional

La incertidumbre que queda sobre X habiendo eliminado toda la incertidumbre sobre Y i.e. conociendo los mensajes de Y

$$I(X=x_i | Y=y_j) = \log \frac{1}{P(X=x_i | Y=y_j)}$$

$$H(X|Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i | Y=y_j)}$$

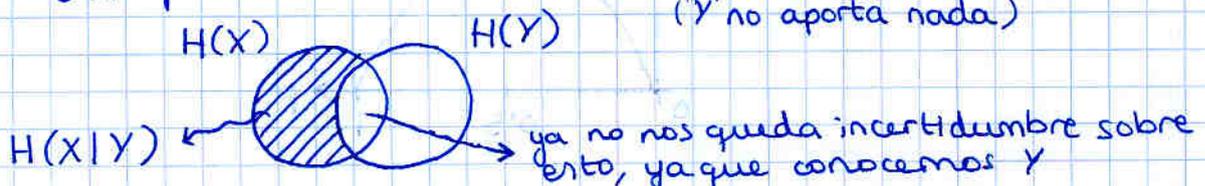
↑
 cuidado, esto no es la prob condicional, ya que sirve para promediar



$$0 \leq H(X|Y) \leq H(X)$$

↑
 Total dependencia entre fuentes

↑
 plena independencia entre fuentes ('Y' no aporta nada)



Ecuaación de Bayes

Se usa mucho en problemas para obtener ciertas probabilidades teniendo otras.

$$P(X=x_i, Y=y_j) = P(X=x_i | Y=y_j) \cdot P(Y=y_j)$$

Se suele escribir despejando $P(X|Y)$, pero me gusta más así

Propiedades de entropía condicional

$$H(X|Y) = H(X, Y) - H(Y)$$

matemáticamente se demuestra sustituyendo en la definición el Teorema de Bayes, aplicando las leyes de los logaritmos y sacando del sumatorio lo que se pueda.

Se ve enseguida con



Inecuación de Fann

- X e Y mismo alfabeto
- definimos como error E cuando $X \neq Y$ (i.e. canal de comunic) error con probabilidad p

$$H(X|Y) \leq H(E) + p \log(N-1)$$

Caso canal simétrico binario

$$H(X|Y) \leq H_2(p)$$

1.6 Información mutua

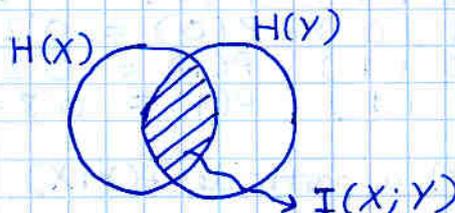
La incertidumbre de X que ha sido liberada al conocer Y. Otra forma de verlo es la incertidumbre que X e Y comparten (i.e. mutua)

$$I(X; Y) = H(X) - H(X|Y)$$

$$0 \leq I(X; Y) \leq H(X)$$

↑
X totalmente independiente de Y

↑
si $X=Y$



Desarrollándolo:

$$I(X; Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log \left(\frac{P(X=x_i | Y=y_j)}{P(X=x_i)} \right)$$

Propiedades

$$I(X; Y) = I(Y; X)$$

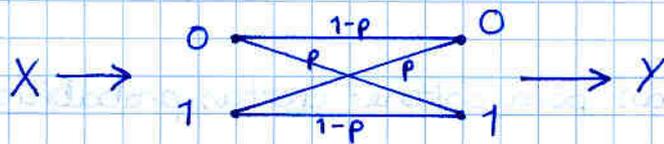
$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

con 3 fuentes:

$$I(Z; X, Y) = H(Z) - H(Z|X, Y)$$

$$I(Z; Y) \leq I(Z; X, Y)$$

Ejemplo: Canal Binario Simétrico (BSC)



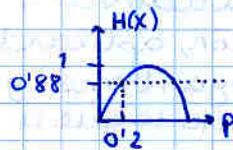
p : probabilidad de error de bit

Datos:

- Canal sin memoria (independiente cada instante)
- $P(X=0) = 0.7$
- $P(X=1) = 0.3$
- $p = 0.2$

a) $H(X)$. Entropía de la fuente

$$H(X) = \sum_{i=0}^1 P(X=x_i) \cdot \log_2 \frac{1}{P(X=x_i)} = 0.7 \cdot \log_2 \frac{1}{0.7} + 0.3 \log_2 \frac{1}{0.3} = 0.88 \text{ bit} \quad \left(\begin{array}{l} 0.88 \text{ bit/simbolo} \\ \text{un poco redundante} \end{array} \right)$$



b) $H(X|Y)$. i.e. Si observamos la salida, ¿qué incertidumbre me queda sobre lo que entró?

$$H(X|Y) = \sum_{i=0}^1 \sum_{j=0}^1 P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i | Y=y_j)}$$

no tenemos ni $P(X=x_i, Y=y_j)$
ni $P(X=x_i | Y=y_j)$

lo que si tenemos es:

$$p = P(Y=0 | X=1) = P(Y=1 | X=0) = 0.2$$

$$1-p = P(Y=0 | X=0) = P(Y=1 | X=1) = 0.8$$

también tenemos $P(X=0) = 0.7$
 $P(X=1) = 0.3$

• obtener lo que nos falta por el teorema de Bayes

$$P(X=x_i, Y=y_j) = P(Y=y_j | X=x_i) \cdot P(X=x_i)$$

$$P(0, 0) = P(0 | 0) \cdot P(0) = 0.8 \cdot 0.7 = 0.56$$

$$P(0, 1) = P(1 | 0) \cdot P(0) = 0.2 \cdot 0.7 = 0.14$$

$$P(1, 0) = P(0 | 1) \cdot P(1) = 0.2 \cdot 0.3 = 0.06$$

$$P(1, 1) = P(1 | 1) \cdot P(1) = 0.8 \cdot 0.3 = 0.24$$

• En cuanto a $P(X|Y)$ a partir de $P(Y|X)$, teorema de Bayes

$$P(X=x_i | Y=y_j) = \frac{P(X=x_i, Y=y_j)}{P(Y=y_j)}$$

- Para calcularlo, aun nos falta $P(Y=y_j)$ que lo calculamos con un resultado no visto en teoria

$$P(Y=y_j) = \sum_{i=0}^{N-1} P(X=x_i, Y=y_j) \quad \textcircled{i}$$

↑ jota
↑ no
↑ coma,
↑ jota (hija)

$$P(Y=0) = P(X=0, Y=0) + P(X=1, Y=0)$$

$$= 0'56 + 0'06 = 0'62$$

$$P(Y=1) = P(X=0, Y=1) + P(X=1, Y=1)$$

$$= 0'14 + 0'24 = 0'38$$

- Ya podemos calcular

$$P(X=x_i | Y=y_j) = \frac{P(X=x_i, Y=y_j)}{P(Y=y_j)}$$

$$P(0 | 0) = \frac{P(0, 0)}{P(0)} = 0'56 / 0'62 = 0'903$$

$$P(0 | 1) = \frac{P(0, 1)}{P(1)} = 0'14 / 0'38 = 0'368$$

$$P(1 | 0) = \frac{P(1, 0)}{P(0)} = 0'06 / 0'62 = 0'097$$

$$P(1 | 1) = \frac{P(1, 1)}{P(1)} = 0'24 / 0'38 = 0'632$$

- Ya disponemos de todo para calcular la entropia conjunta

$$H(X|Y) = 0'56 \log \frac{1}{0'903} + 0'14 \log \frac{1}{0'368}$$

$$+ 0'06 \log \frac{1}{0'097} + 0'24 \log \frac{1}{0'632}$$

$$= 0'64 \text{ bit}$$

c) $I(X; Y) \textcircled{=} = \textcircled{x} - \textcircled{x \cap y}$

$$I(X; Y) = H(X) - H(X|Y)$$

$$I(X; Y) = 0'88 - 0'64 = 0'24 \text{ bit}$$

cuidado:

$P(X|Y)$



$H(X|Y)$



↑ la incertidumbre que nos queda de x que rándonos la de Y

Nos hemos ahorrado 0'24 bit de incertidumbre acerca de la fuente X habiendo visto la salida Y

El número en sí no nos dice nada; la utilidad de esto es poder comparar varios canales distintos

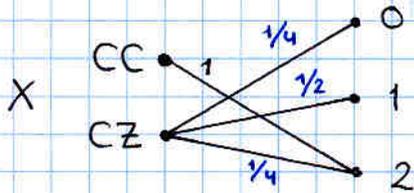
ejemplo: 2 monedas en un saco

- una es falsa: tiene dos caras (CC)
- otra verdadera: cara y cruz equiprobable (CZ)

• se saca una moneda del saco (equiprobablemente) y se tira dos veces, sumando el número de caras.

Llamamos Fuente X a la moneda escogida $X = \{CC, CZ\}$
 x_0, x_1

y Fuente Y al número de caras obtenido $Y = \{0, 1, 2\}$
 y_0, y_1, y_2



$$P(t_1 = Z, t_2 = Z) = 0.25$$

$$P(t_1 = C, t_2 = C) = 0.25$$

$$P(t_1 = Z, t_2 = C) + P(t_1 = C, t_2 = Z) = 0.5$$

a) $H(X)$ $H(X) = 1$ bit (equiprobables)

b) $H(X|Y)$

$$H(X|Y) = \sum_{i=0}^1 \sum_{j=0}^2 P(X=x_i, Y=y_j) \cdot \log_2 \frac{1}{P(X=x_i|Y=y_j)}$$

necesitamos $P(X, Y)$ - (Tª de Bayes)

$$P(X=x_i, Y=y_j) = P(Y=y_j | X=x_i) \cdot P(X=x_i)$$

$$P(X=CC, Y=0) = 0$$

$$P(X=CC, Y=1) = 0$$

$$P(X=CC, Y=2) = 1/2$$

$$P(X=CZ, Y=0) = 1/2 \cdot 1/4 = 1/8$$

$$P(X=CZ, Y=1) = 1/2 \cdot 1/2 = 1/4$$

$$P(X=CZ, Y=2) = 1/2 \cdot 1/4 = 1/8$$

Para calcular

$$P(X=x_i | Y=y_j) = \frac{P(X=x_i, Y=y_j)}{P(Y=y_j)}$$

necesitamos

$$P(Y=y_j) = \sum_i P(X=x_i, Y=y_j)$$

$$P(Y=0) = P(X=CC, Y=0) + P(X=CZ, Y=0) = 0 + 1/2 \cdot 1/4 = 1/8$$

$$P(Y=1) = 0 + 1/2 \cdot 1/2 = 1/4$$

$$P(Y=2) = 1/2 + 1/2 \cdot 1/4 = 5/8$$

Ya podemos calcular $P(X=x_i | Y=y_j) = \frac{P(X=x_i, Y=y_j)}{P(Y=y_j)}$

$$P(X=CC | Y=0) = \frac{P(X=CC, Y=0)}{P(Y=0)} = 0$$

$$P(X=CC | Y=1) = \frac{P(X=CC, Y=1)}{P(Y=1)} = 0$$

$$P(X=CC | Y=2) = \frac{P(X=CC, Y=2)}{P(Y=2)} = \frac{1/2}{5/8} = \frac{4}{5}$$

$$P(X=CZ|Y=0) = \frac{P(X=CZ, Y=0)}{P(Y=0)} = \frac{1/8}{1/8} = 1 \quad \leftarrow \text{no te sorprendas. Piensalo } \rightarrow \text{ en correcto}$$

$$P(X=CZ|Y=1) = \frac{P(X=CZ, Y=1)}{P(Y=1)} = \frac{1/4}{1/4} = 1$$

$$P(X=CZ|Y=2) = \frac{P(X=CZ, Y=2)}{P(Y=2)} = \frac{1/8}{5/8} = \frac{1}{5}$$

Ya lo tenemos todo para calcular $H(X|Y)$

$$H(X|Y) = 0 \log \frac{1}{0} + 0 \log \frac{1}{0} + \frac{1}{2} \log \frac{1}{4/5}$$

$$+ \frac{1}{8} \log \frac{1}{1} + \frac{1}{4} \log \frac{1}{1} + \frac{1}{8} \log \frac{1}{1/5}$$

$$= 0'451 \text{ bit}$$

c) $I(X; Y)$

$$I(X; Y) = H(X) - H(X|Y)$$

$$= 1 - 0'451$$

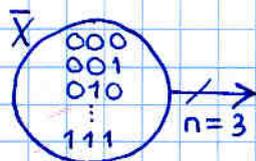
$$= 0'55 \text{ bit}$$

ejercicio: hacerlo con 3 tiradas
Res: $I(X, Y) = 0'717 \text{ bit}$

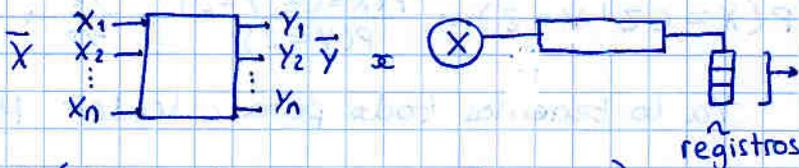
Veremos que $I(X; Y)$ aumenta cuantas mas tiradas; es lógico.

1.7 Vectores Aleatorios Discretos

$\vec{X} = (X_1, X_2, X_3, \dots, X_n)$ ej. tx. de tramas de n bits



Puede imaginarse como un canal paralelo o un canal serie sin memoria



$$P(\vec{X} = \vec{x}) = P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$$

Si las X_n son independientes:

$$P(\vec{X} = \vec{x}) = \prod_{i=1}^n P(X_i = x_i)$$

$$P(\vec{Y} = \vec{y} | \vec{X} = \vec{x}) = \prod_{i=1}^n P(Y_i = y_i | X_i = x_i)$$

y se puede demostrar:

$$I(\vec{X}, \vec{Y}) = \sum I(X_i = x_i, Y_i = y_i)$$

1.8 Teorema de Proceso de la Información

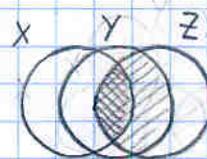
Secuencia o
Cadena
de Markov



todo lo que X
influya en Z
viene ya reflejado
en Y .

$$P(Z = z_i | Y = y_j, X = x_k) = P(Z = z_i | Y = y_j)$$

$$\Leftrightarrow I(Z; X, Y) = I(Z; Y)$$



$$I(X; Y, Z) = I(X; Y)$$

- Intuitivamente; la información mutua entre un punto de la cadena y otros ciertos puntos es igual a la que tiene entre el punto más cercano de esos 'ciertos puntos'.

Ademas, aplicando que $I(X; Z) \leq I(X; Z, Y)$

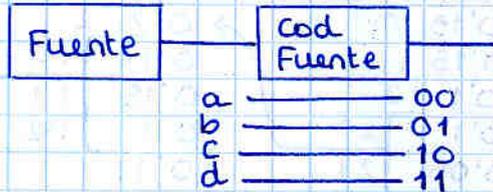
$$I(X; Z) \leq \begin{cases} I(X; Y) \\ I(Y; Z) \end{cases}$$

- Intuitivamente: Cada bloque limita la información mutua. El bloque de menor información mutua entre su entrada y salida limita a toda la cadena.
- Por muchos bloques que pongamos para mejorar I , en el mejor caso la dejariamos igual.



2. Fuentes Discretas y Codificación de Fuente

Codificación Típica



$$\bar{n} = 2 \text{ bits/símbolo de fuente}$$

¿Existe alguna correspondencia que minimice el nº de bits a enviar?
 Si: algoritmo de Huffman y extensión de fuente

ej
 a (50%) → 0
 b (30%) → 10
 c (15%) → 110
 d (5%) → 111

$$\bar{n} = 0.5 \cdot 1 + 0.3 \cdot 2 + 0.2 \cdot 3 = 1.7 \text{ bit/símbolo de fuente}$$

¿Cual sería el valor mínimo?

Teorema de codificación de fuente de Shannon

$$\bar{n} \geq H(X)$$

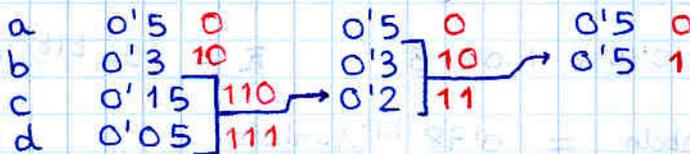
en nuestro ejemplo:
 $H(X) = 0.5 \cdot \log_2 \frac{1}{0.5} + 0.3 \cdot \log_2 \frac{1}{0.3} + 0.15 \cdot \log_2 \frac{1}{0.15} + 0.05 \cdot \log_2 \frac{1}{0.05} = 1.6477 \text{ bit}$

2.4. Algoritmo de Huffman

Ordenar símbolos de fuente según probabilidad

Ir agrupando de 2 en 2 y ordenando

Una vez se llega a tantas probabilidades como símbolos, se asignan los símbolos:



- En general, con código R-ario (en nuestro ejemplo binario → R=2) se van agrupando las R probabilidades más bajas
- Excepto la primera vez / primera columna en la que se deben agrupar [2, R] probabilidades para que en el paso final nos queden R probabilidades para asignar R símbolos

Debe cumplirse $R^k \equiv 1 \pmod{R-1}$ (simbolos de fuente el RESTO de la división)

$$R^k \equiv 1 \pmod{R-1} = N \pmod{R-1}$$

↪ la agrupación de la primera columna (entre 2 y R)

ejemplo $R=3, N=8$

La agrupación de la primera columna será o dos o tres.

$$R^{k=1} \text{ MOD } (R-1) = N \text{ mod } (R-1)$$

$R^{k=1}=3$		3	MOD	2	\neq	8	MOD	2
$R^{k=2}=2$		2	MOD	2	$=$	8	MOD	2 = 0 \checkmark

a	0'4	0	→	0'4	0	→	0'4	0	→	0'4	0
b	0'15	10	→	0'15	10	→	0'2	2	→	0'4	1
c	0'15	11	→	0'15	11	→	0'15	10	→	0'2	2
d	0'1	12	→	0'1	12	→	0'15	11			
e	0'1	20	→	0'1	20	→	0'1	12			
f	0'05	21	→	0'05	21						
g	0'04	220	→	0'05	22						
h	0'01	221	→								

2.5 Extensión de Fuente

ejemplo X_1 $\begin{matrix} a \\ b \end{matrix}$ $\begin{matrix} P(X_1=a)=0'8 \\ P(X_1=b)=0'2 \end{matrix}$

• Algoritmo de Huffman: $\begin{matrix} a & 0'8 & 0 \\ b & 0'2 & 1 \end{matrix}$ $\bar{n} = 1 \text{ bit/símbolo de } X_1$

• El límite mínimo es: $H(X_1) = \dots = 0'7219 \text{ bit/símbolo de } X_1$

• Realicemos Extensión de Fuente:



Algoritmo de Huffman

aa	0'64	0	→	0'64	0	→	0'64	0
ab	0'16	11	→	0'2	10	→	0'36	1
ba	0'16	100	→	0'16	11			
bb	0'04	101						

$$\bar{n} = 0'64 \cdot 1 + 0'16 \cdot 2 + 0'2 \cdot 3 \quad \bar{n} = 1'56 \text{ bit/símbolo de } X_2$$

Pero CUIDADO $1'56 \text{ bit/símbolo de } X_2 = 0'78 \text{ bit/símbolo de } X_1$

cuantos más símbolos agrupemos para hacer extensión de fuente, más acercamos \bar{n} a su límite mínimo $H(X)$

ej. Código Repetición

$k=1$ $n=100 \Rightarrow$ Por cada bit lo repetimos 100 veces

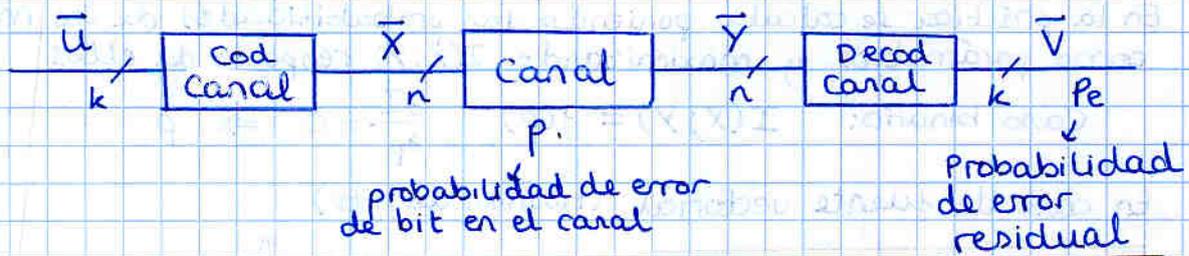
- El receptor decide el bit según mayoría.
- Sólo nos equivocariamos si hubiesen > 50 errores.
- Además cuanto mayor fuese n , más seguro sería el sistema

Por tanto: Interesa R bajo? Sin embargo un R muy bajo es obvio que malgasta ancho de banda.

$$R = \frac{k}{n} \text{ bajo} \Rightarrow \begin{array}{l} \text{mucha seguridad} \\ \text{mucho gasto de ancho de banda} \end{array}$$

Interesa R alto para no gastar BW pero lo suficientemente bajo como para tener cierta seguridad, o tanta seguridad como queramos.

3.1. Teorema de codificación de canal de Shannon



Demostración:

$$I(\bar{U}, \bar{V}) = \sum_{i=1}^k I(U_i, V_i) = \sum_{i=1}^k H(U_i) - H(U_i|V_i) \leq I_{\max}(\bar{U}, \bar{V})$$

por un lado:

$$\sum_{i=1}^k 1 - H(U_i|V_i) \leq \sum_{i=1}^k H(U_i) - H(U_i|V_i)$$

ya que $H(U_i) \leq 1$ (fuente binaria)

por otro lado

$$I_{\max}(\bar{U}, \bar{V}) \leq I_{\max}(\bar{X}, \bar{Y}) = Cn$$

Teorema
Proceso de
Información

por tanto:

$$\Rightarrow \sum_{i=1}^k 1 - H(U_i|V_i) \leq Cn$$

por un lado

$$H(U_i|V_i) \leq H_2(P_e) + P_e \log(2-1) \leq H_2(P_e)$$

Desigualdad de Fann

entropía de la fuente binaria 'suceso error'

suceso error coincide con prob residual de error

por otro lado

como $C = \sup_{\forall n} \frac{1}{n} Cn$

$$Cn \leq n \cdot C$$

por tanto:

$$k \cdot (1 - H_2(P_e)) \leq n \cdot C$$

$$\Rightarrow R = \frac{k}{n} \leq \frac{C}{1 - H_2(P_e)}$$

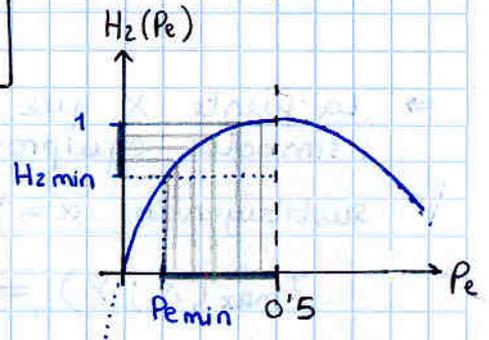
Teorema de Codificación de canal de Shannon

$$R = \frac{R}{n} \leq \frac{C}{1 - H_2(P_e)}$$

Despejando:

$$H_2(P_e) \geq 1 - \frac{C}{R}$$

Un límite mínimo de H_2 nos establece un LÍMITE MÍNIMO DE P_e SEGUN C/R



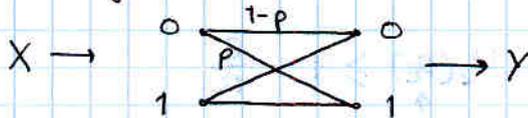
- Si $R > C \Rightarrow H_2(P_e) > 0$
 \hookrightarrow No podemos tener un P_e tan pequeño como queramos

- Si $R < C \Rightarrow H_2(P_e) > -\infty$
 $\hookrightarrow P_e$ puede estar tan cercano a 0 como queramos

- \Rightarrow
- Interesa R grande para ahorrar ancho de banda
 - Interesa $R \leq C$ para poder hacer P_e tan pequeño como queramos

Solución Ideal sería $R = C \Rightarrow H_2(P_e) \geq 0 \Rightarrow P_e \geq 0$
 Si supieramos aprovechar muy bien los bits de redundancia.
 se está estudiando.
 Actualmente se consiguen $R \leq C$

Ejercicio: BSC



Datos:

$$\begin{aligned} P(X=0) &= 0.7 \\ P(X=1) &= 0.3 \\ p &= 0.2 \end{aligned}$$

- Calcular:
- $H(X) = 0.88$ bit
 - $H(X|Y) = 0.64$ bit
 - $I(X;Y) = 0.24$ bit
 - Capacidad C

Hay que calcular I tomando parámetros de las probabilidades de X
 $P(X=0) = \alpha \quad P(X=1) = 1 - \alpha$

$$I(X;Y) = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log \frac{P(X=x_i|Y=y_j)}{P(X=x_i)} = \sum_i \sum_j P(X=x_i, Y=y_j) \cdot \log \frac{P(X=x_i, Y=y_j)}{P(X=x_i) \cdot P(Y=y_j)}$$

Necesitamos:

$$P(X=x_i, Y=y_j) = P(Y=y_j|X=x_i) \cdot P(X=x_i)$$

$$P(0,0) = P(0|0) \cdot P(0) = (1-p)\alpha$$

$$P(0,1) = \dots = p \cdot \alpha$$

$$P(1,0) = \dots = p \cdot (1-\alpha)$$

$$P(1,1) = \dots = (1-p)(1-\alpha)$$

y Necesitamos

$$P(Y=y_j) = \sum_i P(X=x_i, Y=y_j)$$

$$P(Y=0) = (1-p) \cdot \alpha + p \cdot (1-\alpha)$$

$$P(Y=1) = p \cdot \alpha + (1-p)(1-\alpha)$$

Ya podemos calcular $I(X;Y)$

Sustituyendo: $I(X;Y) = f(p, \alpha)$

Haciendo: $\frac{\partial I(X;Y)}{\partial \alpha} = 0 \Rightarrow (\dots\dots) \Rightarrow$ despejando

$$\alpha = \frac{1}{2}$$

Esto es cierto para todos los BSC

\Rightarrow La fuente X que 'maximiza' $I(X;Y)$ es la que tiene símbolos equiprobables

sustituyendo $\alpha = 1/2$ en $I(X;Y)$

$$\begin{aligned} I_{\max}(X;Y) &= 1 + (1-p) \cdot \log_2(1-p) + p \log_2 p \\ &= - \left[(1-p) \log_2 \frac{1}{1-p} + p \log_2 \frac{1}{p} \right] \\ &= - H_2(p) \end{aligned}$$

$$I_{\max}(X;Y) = \boxed{1 - H_2(p) = C} \quad \text{Fórmula general para BSC}$$

en el BSC con $p=0.2$



¿Podemos conseguir $P_e < 0.02$?

Tª Codificación de Shannon: $R \leq \frac{C}{1 - H_2(P_e)} \rightarrow H_2(P_e) \geq 1 - \frac{C}{R}$

$$C = 1 - H_2(p) = 1 - H_2(0.2) = 0.278 \text{ bit}$$

prob error por bit del canal

$$R = \frac{k}{n} = \frac{1}{3}$$

$$H_2(P_e) \geq 1 - \frac{C}{R} = 1 - \frac{0.278}{1/3} = 0.1652$$

$H_2(P_e) \geq 0.1652$ fija una P_e mínima que podríamos hallar por prueba y error. Pero como lo que piden es si podemos lograr $P_e < 0.02$:

$$H_2(0.02) = 0.02 \log_2 \frac{1}{0.02} + 0.98 \log_2 \frac{1}{0.98} = 0.1415 \text{ bit} < 0.1652$$

\Rightarrow No podemos conseguir $P_e < 0.02$

Problemas de Examen

1. X V.A. de entrada Canal BSC Y V.A. de salida
 $p = 0.25$

- a. Siempre $I(X; Y) = 0.199$ bit } $I(X; Y) = H(X) - H(X|Y)$ no nos dan datos.
 b. Siempre $I(X; Y) = 0.811$ bit }
 c. La capacidad $C = 1$ bit } $C = 1 - H_2(p)$ $H_2(p) \neq 0$ $I(X; Y)$ depende de $H(X)$
 → d. Ninguna es correcta

2. Sea X V.A. de entrada Canal BSC Y V.A. de salida
 $p = 0.5$

- a. Siempre $H_2(Y) = 1$ bit
 b. Siempre $H_2(X) = H_2(Y) = 1$ bit → no tenemos datos de $H_2(X)$
 c. La capacidad del canal $C = 1$ bit → $C = 1 - H_2(p) = 0$ bit
 d. Ninguna es correcta el canal no tiene capacidad alguna para transmitir

$H(Y)$:

$$P(Y=0) = P(X=0, Y=0) + P(X=1, Y=0)$$

$$P(Y=1) = P(X=0, Y=1) + P(X=1, Y=1)$$

$$P(X=0, Y=0) = P(Y=0|X=0) \cdot P(X=0) = (1-p) \cdot \alpha$$

$$P(X=0, Y=1) = p \cdot \alpha$$

$$P(X=1, Y=0) = p \cdot (1-\alpha)$$

$$P(X=1, Y=1) = (1-p) \cdot (1-\alpha)$$

$$P(Y=0) = (1-p)\alpha + p(1-\alpha) = \alpha + p - 2p\alpha = \alpha + \frac{1}{2} - \alpha = \frac{1}{2}$$

$$P(Y=1) = \frac{1}{2}$$

3. Se emplea el algoritmo de Huffman con la fuente X

$$P(X=0) = 0.49$$

$$P(X=3) = 0.04$$

$$P(X=6) = 0.14$$

$$P(X=1) = 0.01$$

$$P(X=4) = 0.07$$

$$P(X=7) = 0.02$$

$$P(X=2) = 0.07$$

$$P(X=5) = 0.14$$

$$P(X=8) = 0.02$$

El código tiene una longitud media de l dígitos binarios/símbolo

- a. Según el teorema de codificación de fuente $l \leq H_2(X)$
 b. Cualquier código con 9 palabras código de longitudes $\{6; 6; 5; 4; 4; 4; 3; 3; 1\}$ es siempre código de Huffman
 c. $l = 1.1567$ dígitos binarios/símbolo
 d. Ninguna es correcta

$$H_2(X) = \sum_{i=0}^8 P(X=x_i) \cdot \log_2 \frac{1}{P(X=x_i)}$$

$$\log_8 z = \frac{\log_2 z}{\log_2 8}$$

$$= \sum P(X=x_i) \cdot \frac{1}{\log_2 8} \cdot \log_2 \frac{1}{P(X=x_i)}$$

$$\rightarrow H_8(X) = \frac{1}{3} H_2(X)$$

a) $l = \bar{n} \geq H_2(X) = 3H_8(X)$

Teorema de codificación de fuente

b) No tiene porque

c) Hay que calcular ℓ para saber si es c) o d)

4. Fuente variable aleatoria X

$$P(X=0) = 0'15$$

$$P(X=3) = 0'01$$

$$P(X=2) = 0'02$$

$$P(X=1) = 0'2$$

$$P(X=20) = 0'45$$

$$P(X=-2) = 0'08$$

$$P(X=6) = 0'01$$

$$P(X=-1) = 0'08$$

se emplea código de Huffman ternario

Código tiene longitud media de ℓ dígitos ternarios/símbolo

a. Según teorema de codificación de fuente $\ell \geq H_3(X) = 2'22$

b. $\ell = 1'49$ dígitos ternarios/símbolo

c. $\{0; 1; 20; 21; 211; 2220; 2221\}$ es un posible conjunto de palabras código

d. Ninguna es correcta

a. En efecto se cumple

$$\ell \geq H_3(X)$$

↑
dígitos ternarios/símbolo

↑
entropía ternaria

Hay que calcular $H_3(X)$

b. Hacer código Huffman y obtener ℓ

c. No, no es unívocamente decodificable \rightarrow no es Huffman

5. Tamaño mínimo de fichero de 2kbyte de texto ASCII, caracteres independientes de 8 bits, cuya redundancia es de 3 bits/byte

La redundancia es la diferencia entre los bits que tendría por carácter si fueran equiprobables y los bits por carácter en el caso óptimo en el cual es igual a la redundancia de la fuente.

$$8 - H(X) = 3 \rightarrow H(X) = 5 \text{ bits/carácter}$$

En ese caso óptimo el tamaño será:

$$\frac{2 \text{ kbyte}}{8} \cdot 5 \text{ bits} = 1'25 \text{ kbyte}$$

↑
nº de caracteres

6. En canal binario simétrico con alta capacidad...

a. se obtendrá probabilidad de error cercana a $\frac{1}{2}$, mal -

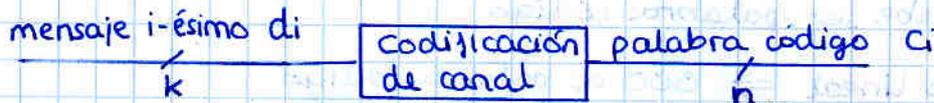
\rightarrow b. " " " " " " " a 0

c. la prob err disminuye según $C > 1$. mal en BSC $C \leq 1$

$$P_e = \frac{1}{2} \text{ a } 0$$

TEMA 3. Control de Errores

1. Control de errores



$$d_i = [d_{i0} d_{i1} \dots d_{i_{k-1}}]$$

$$k < n$$

$$C_i = [C_{i0} C_{i1} \dots C_{i_{n-1}}]$$

los mensajes de k en k bits independientemente de las palabras código de codificación de fuente que lleguen.

si $k=n$:
 $R=1 \rightarrow C=1$
 canal ideal

$C(n, k)$: conjunto de palabras código

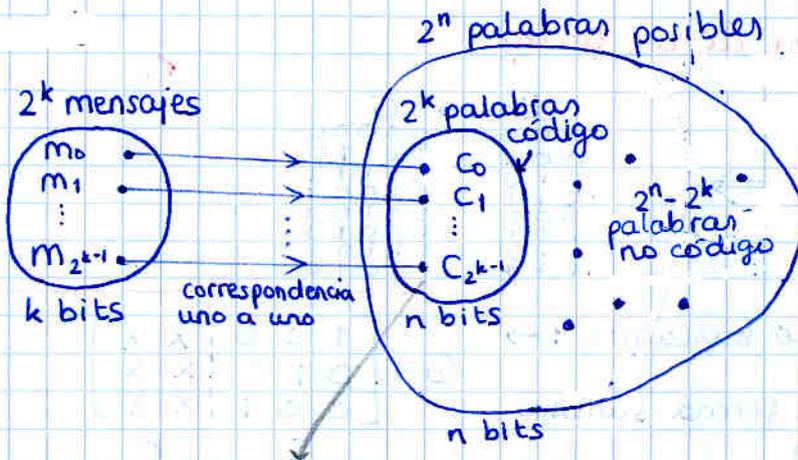
Características:

- Código Bloque: la C_i sólo depende de la d_i (mensaje ACTUAL)

Operaciones

Suma = Resta = XOR = \oplus
 = '0' si iguales, '1' si bits distintos

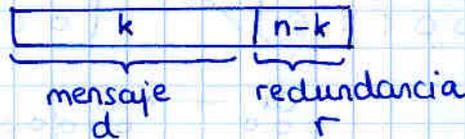
$$\oplus \begin{array}{r} 10011 \\ 11001 \\ \hline 01010 \end{array}$$



Cuando en el receptor se recibe una palabra no código:

- ARQ: automatic repetition request
- FEC: forward error correction. Comunica a niveles superiores la pal. código más cercana a lo recibido

- Código Lineal: Cualquier suma de 2 palabras código es palabra código
- Código sistemático: el mensaje se incluye en la palabra código

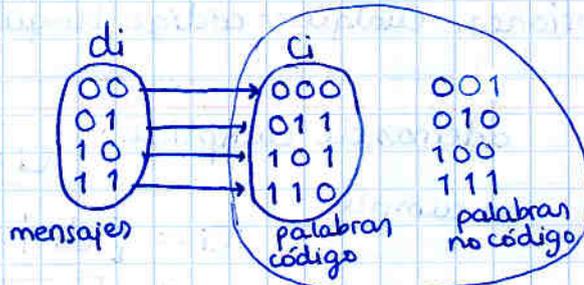


$$c = [d | r]$$

ejemplo: código $C(3,2)$ con paridad par

- la suma de bits del mensaje es igual al bit de paridad
- la suma de bits de la palabra código es cero (nº par de unos se anulan)

$$\begin{aligned} C_{i0} &= d_{i0} \\ C_{i1} &= d_{i1} \\ C_{i2} &= d_{i0} \oplus d_{i1} \end{aligned}$$



- código bloque
- sistemático
- lineal

TRUCO: Para descartar que un código es lineal

- La suma de cualquier palabra consigo misma es 000
- Si es lineal, la suma de una palabra código con cualquier otra debe ser palabra código

⇒ 000 debe ser palabra código

Código lineal ⇒ 000 es palabra código

2. Códigos Bloque Lineales

2.1 Matriz Generadora

$$C_i = d_i \cdot G$$

$1 \times n$ $1 \times k$ $k \times n$

k filas linealmente independientes
 ↳ subespacio k -dimensional
 ↳ en binario, cada dimensión añade la posibilidad 0 o 1 extra
 ⇒ 2^k vectores diferentes

las k filas son siempre palabras código

Sistematización de un código

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \Rightarrow \begin{array}{l} d_i \rightarrow C_i \\ \begin{matrix} 00000 \\ 00001 \\ 00100 \\ 01000 \\ 10000 \\ 11000 \\ 11001 \end{matrix} \end{array}$$

Para que sea sistemático buscamos: →

Para llevar a G a esa forma sumamos sus filas

$$G = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & X & X \\ 0 & 1 & 0 & X & X \\ 0 & 0 & 1 & X & X \end{array} \right]$$

I P
 matriz matriz
 identidad de paridad

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{\oplus} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \xrightarrow{\oplus} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$G = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{array} \right] \Rightarrow \begin{array}{l} d_i \rightarrow C_i \\ \begin{matrix} 00000 \\ 00001 \\ 00100 \\ 01000 \\ 10000 \\ 11000 \\ 11001 \end{matrix} \end{array}$$

Las palabras código son las mismas.
→ misma potencia antierror

No hay códigos sistemáticos/no sistemáticos. Son los mismos códigos con distintas versiones. Cualquier código bloque lineal se puede sistematizar

Si:

$$G = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & X & X \\ 0 & 1 & 0 & X & X \\ 0 & 0 & 1 & X & X \end{array} \right]$$

I P

además de cumplirse:

$$C_i = d_i \cdot G$$

se cumple

$$C_i = [d_i \mid r_i] \quad \text{por ser sistemático (} r_i: \text{ bits paridad)}$$

siendo

$$r_i = d_i \cdot P \quad \text{matriz de paridad}$$

mensaje

Comprobación de Paridad

1. comprobando palabra recibida con una lista de palabras código (ineficiente)
2. Si es sistemático: cogemos el mensaje
Recalcular paridad:
comprobamos si $r = d \cdot P$
(aún es 'muy a mano')

$$Cr = \underbrace{[d]}_{\text{mensaje}} \mid r$$

El mejor método si el código es sistemático

3. Matriz de comprobación de paridad

$$G = \begin{bmatrix} 1 & 0 & 0 & | & X_1 & X_2 \\ 0 & 1 & 0 & | & X_3 & X_4 \\ 0 & 0 & 1 & | & X_5 & X_6 \end{bmatrix}$$

$$G = [I_k \mid P]$$

Definimos

$$H^T = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \\ X_5 & X_6 \\ \hline 1 & 0 \\ 0 & 1 \end{bmatrix}$$

matriz de comprobación de paridad

$$H^T = \begin{bmatrix} P \\ I_{n-k} \end{bmatrix}$$

Síndrome s

Receptor recibe la palabra x (n dígitos)

Síndrome de x

$$s = x \cdot H^T$$

$1 \times (n-k)$ $1 \times n$ $n \times (n-k)$

el síndrome es un vector de $n-k$ dígitos
NO UN NUMERO

- Si $x = C_i = (d \mid r)$ palabra código sistemática

$$s = C_i \cdot H^T = 0$$

El síndrome de una palabra código es cero

Dem:

$$(d \mid r) \cdot \begin{bmatrix} P \\ I \end{bmatrix}$$

$$= \underbrace{(d \cdot P)}_{\text{paridad recalculada}} \oplus \underbrace{(r \cdot I)}_{\text{paridad recibida}}$$

$$= \vec{0} \text{ si son iguales}$$

- Si x es una palabra código que ha sufrido un error

$$x = C_i + e$$

e patrón de error introducido por el canal (un 1 en los bits erróneos)

$$s = (C_i + e) \cdot H^T = e \cdot H^T$$

El síndrome de una palabra código con un error es igual al síndrome del error

Vemos que esta herramienta puede llegar a ser muy poderosa, no sólo para detectar errores, sino para saber cual ha sido el error y corregirlo. Gran enemigo: si e es una palabra código

Sería óptimo que los errores más probables consiguiéramos que tuviesen síndromes distintos

como $s = x \cdot H^T \Rightarrow$ Hay 2^{n-k} síndromes distintos

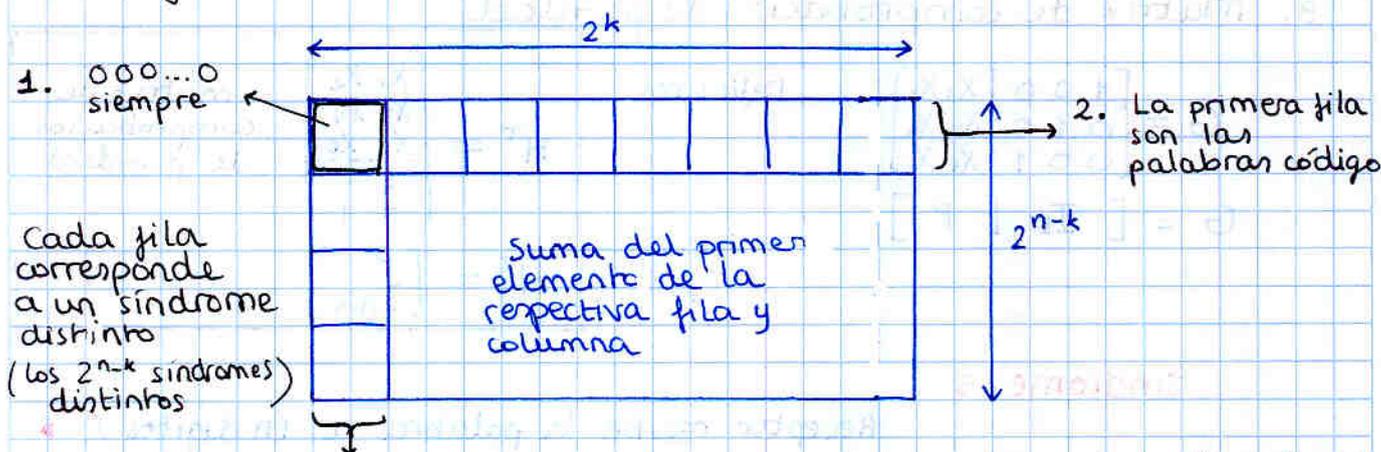
$1 \times (n-k)$ $1 \times n$ $n \times (n-k)$

Tabla estándar

Nos permite ver varios detalles de una manera simple

- Si dos patrones de error equiprobables (i.e. tienen el mismo número de unos) tienen el mismo síndrome, entonces son incorregibles: si el receptor calcula dicho síndrome no puede decantarse por un error u otro
- La tabla estándar nos permite tener organizados los patrones de error según su síndrome (una fila para cada síndrome)

Confección de la tabla:



- La 1ª columna la vamos rellendo fila a fila. Son los patrones de error más probables. Pondremos en la 2ª fila el patrón de error más probable (i.e. cualquiera de los que tengan sólo un uno) Rellenamos toda la fila sumando ese patrón de error a las pal. código. En la siguiente fila ponemos el siguiente patrón de error más probable (i.e. será otro de sólo un uno) siempre y cuando NO APAREZCA ya en lo que llevamos de tabla (los elementos no se repiten) Así vamos rellorando las filas con los patrones de error más probables (1º los que quepan de un sólo uno; si se nos acaban, los de dos unos, y así sucesivamente hasta llenarlo todo)

ej:

$$G = \begin{bmatrix} 1 & 0 & 0 & | & 1 & 1 \\ 0 & 1 & 0 & | & 0 & 1 \\ 0 & 0 & 1 & | & 1 & 0 \end{bmatrix}$$

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

síndrome:

00	→	00000	00110	01001	01111	10011	10101	11010	11100
11	→	10000	10110	11001	11111	00011	00101	01010	01100
01	→	00001	00111	01000	01110	10010	10100	11011	11101
10	→	00010	00100	01011	01101	10001	10111	11000	11110

La tabla permite saber si ciertas palabras recibidas se pueden corregir por máxima verosimilitud i.e. suponiendo siempre que el error ha sido el de menos bits (el más probable) y corrigiéndolo.

- Si en una fila no hay ninguna palabra con el mismo número de unos que la de la primera columna, el patrón de error es CORREGIBLE. Al recibir cualquiera de esas palabras el receptor y calcular su síndrome, supondrá que el error ha sido el de la primera columna (el único con ese síndrome que es tan probable) Pasará al nivel superior la palabra código de esa misma columna.

- Si en una fila sí que hay una palabra igual de probable que la de la 1ª columna, entonces el error es DETECTABLE (pero no corregible) ya que si el receptor calcula el síndrome de lo recibido no sabrá por cual de los dos errores equiprobables decantarse.

En el ejemplo puedes comprobar que la 2ª fila son todas palabras/errores (en la tabla algunas cosas se interpretan como palabras/errores indistintamente) corregibles, y la 3ª y 4ª fila son tan sólo detectables: ej 00111 se parece tanto a 00110 como a 01111.

NOTA: Tal vez podría decantarse si uno de los mensajes fuera más probable que el otro, pero eso no se llama máxima verosimilitud

Todo ello se puede implementar en el receptor con una tablita de los síndromes, si es o no corregible, y su patrón de error (en caso de que lo sea)

Los códigos cuyos síndromes están todos asociados a patrones corregibles (i.e. tienen en la 1ª columna todos los errores de 1 bit) se llaman **CODIGOS PERFECTOS**: sólo existen 3 en el mundo.

A pesar del nombre, su característica no les confiere gran potencia antierror.

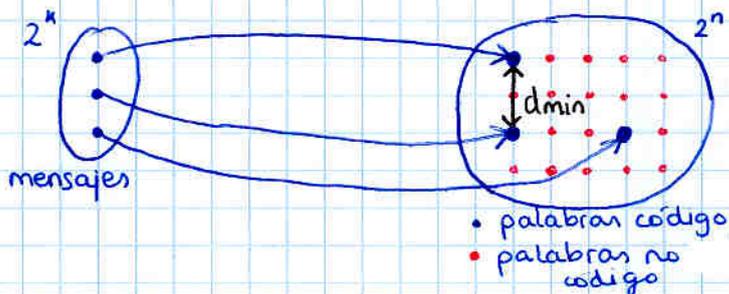
Distancia mínima d_{min}

- Peso Hamming $w(x)$: número de unos que tiene la palabra x
- Distancia Hamming $d(x,y)$: número de bits distintos entre x e y .

Se puede comprobar $d(x,y) = w(x+y)$

- Distancia mínima de un código $C(n,k)$

Distancia mínima entre las palabras código



Conviene que la d_{min} sea lo mayor posible para que los errores del canal resulten con más probabilidad en palabras no código

- Cálculo de d_{min}

1. Mirando distancias entre todas las parejas posibles de palabras código
2. Ya que $d(x,y) = w(x+y)$, y en un código lineal $x+y = z$ (siendo z palabra código)

↳ $d_{min} = \text{peso mínimo de las palabras código } (\neq \bar{0})$

3. Ya que $C_i \cdot H^T = (0 \ 0 \ \dots \ 0)$ siendo en realidad esto el resultado de sumar elemento a elemento las filas de H^T correspondientes a los unos de C_i

d_{\min} = cantidad mínima de filas de H^T que debo sumar elemento a elemento para obtener el vector nulo

1

$$H^T = \begin{pmatrix} 11 \\ 01 \\ 10 \\ 10 \\ 01 \end{pmatrix}$$

$$\oplus \begin{matrix} 01 \\ 01 \\ 00 \end{matrix}$$

$$\rightarrow d_{\min} = 2$$

corresponde a la $C_i = (01001)$

- Si hay 2 filas iguales: $d_{\min} = 2$
- Si existe la fila (00): $d_{\min} = 1$

Capacidad detectora y correctora de un código $C(n,k)$ según d_{\min}

Si representamos las dos palabras código más cercanas se ve:



Un código con d_{\min} :

- detecta todos los errores de $e = d_{\min} - 1$ dígitos binarios o menos
- corrige todos los errores de $t = \lfloor \frac{d_{\min} - 1}{2} \rfloor$ dígitos binarios o menos
- Todos los patrones de error con t unos o menos (i.e. corregibles) tienen un síndrome único diferente de otros patrones con mismos unos

cuidado: errores de $\leq d_{\min} - 1$ dig binarios \Rightarrow detectable
errores de $\leq \lfloor \frac{d_{\min} - 1}{2} \rfloor$ dig binarios \Rightarrow corregible

redondear hacia abajo

3.2.1 Versión sistemática del código

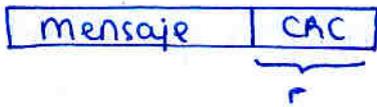
Los códigos generados con $c(x) = m(x) \cdot g(x)$

$$c = m \cdot G_{NS} \leftarrow \text{son no-sistemáticos}$$

Como sabemos, todo código bloque lineal se puede sistematizar, y no cambia el conjunto de palabras código

	generación con $g(x)$	gen con matriz	se cumple
No Sistemático	$c(x) = m(x) \cdot g(x)$	$c = m \cdot G_{NS}$ sistematizar matriz generadora	$m(x) = \frac{c(x)}{g(x)}$ $\text{resto} \left(\frac{c(x)}{g(x)} \right) = 0$ $G_{NS} = \begin{pmatrix} x^{n-1} & g(x) \end{pmatrix}$
Sistemático	$r(x) = \text{resto} \left(\frac{x^{n-k} \cdot m(x)}{g(x)} \right)$ $c(x) = \underbrace{x^{n-k} \cdot m(x)}_{\text{bits CRC}} + r(x)$	$c = m \cdot G_S$ $c = [m \mid r]$	$\text{resto} \left(\frac{c(x)}{g(x)} \right) = 0$ puesto que el conjunto $c(x)$ es el mismo. Aunque $m(x) \neq \frac{c(x)}{g(x)}$

CRC. Cyclic Redundancy Check



$$r(x) = \text{resto} \left(\frac{x^{n-k} \cdot m(x)}{g(x)} \right)$$

el grado del resto es $<$ el grado del divisor

Los polinomios $g(x)$ están estandarizados.

resumen grados:

$$\begin{aligned} g(x) \text{ grado } n-k &\Rightarrow n-k+1 \text{ bits} \\ \text{como grado resto } < \text{ grado divisor} \\ r(x) \text{ grado } n-k-1 &\Rightarrow n-k \text{ bits} \end{aligned}$$

i.e. $\text{gr}(r(x)) \leq n-k-1$

$r: n-k \text{ bits}$

3.3 Algunas propiedades

• Detectan todos los errores simples (i.e. $d_{min} > 1$) \rightarrow (detectan, no corrigen!)

• Ráfagas de error: secuencia de bits en la cual primer y ultimo bits son erróneos (los intermedios pueden o no serlo)

- Se detectan todas ráfagas longitud $\leq n-k$

- Probabilidad de no detectar ráfaga long = $n-k+1$ es $P_{nd} = \frac{1}{2^{n-k-1}}$

- Prob. de no detectar ráfaga long $\geq n-k+2$ $P_{nd} = \frac{1}{2^{n-k}}$

num. de ráfagas de long $n-k-1$ que dan lugar a palabras código

num total ráfagas long $n-k-1$

Además:

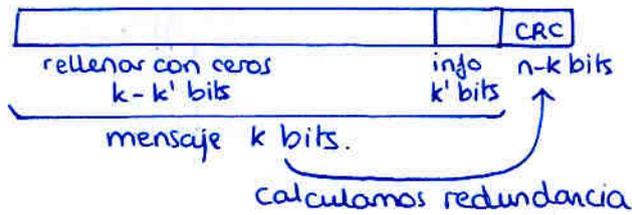
• si $g(x)$ tiene como factor $x^h + 1$ ($h > 0$) \Rightarrow se detectan todos los errores impares (pero hamming = n° bits erróneos impar)

• si $g(x)$ tiene como factor un polinomio primitivo $p(x)$ grado $(p(x)) = r$ $n \leq 2^r - 1$ \Rightarrow se detectan todos los errores dobles

3.4 Variantes de los códigos cíclicos

códigos recortados:

- Podemos conseguir buenos códigos (alta d_{min}) si tomamos n y k altos ej ($k=64K$) aun siendo $n-k$ pequeños.
- Para no tener que esperar a un mensaje de k bits, podemos rellenar con ceros
- si siempre rellenamos con ceros \rightarrow no los enviamos.



el receptor añadirá los ceros para hacer la comprobación de redundancia

Paridad invertida

- El problema con que sean cíclicos es que si se pierde el sincronismo aunque sea un bit, es desastroso.

Es muy facil confundir

con $\begin{array}{ccccccc|ccc} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{array}$

Por ej, envias

$\boxed{1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ X\dots}$

si el sincronismo se va 1 bit

si $X=0$, se confunde lo recibido con una palabra código

Probabilidad 50% error no detectable

- Solución: invertir los bits de paridad
Las palabras código ya no son físicamente "cíclicas" aunque todas las propiedades son las mismas.
El detector invierte los bits de paridad

pal. código $1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1$

tx $\dots \boxed{1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ X\dots}$

rx $\boxed{0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ X}$

error de sincronismo

pal. código interpretada $0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ X$

pal. código con la q era facil confundirse (la original desplazada 1 bit) $0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0$

será equivalente a un error de 1 bit o 2 bits (segun X)

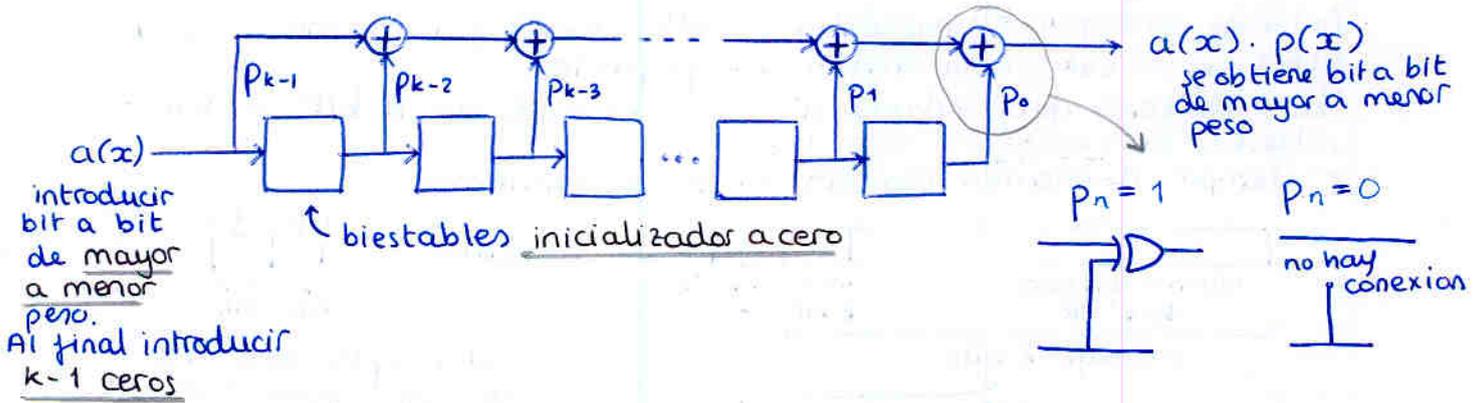
si exigimos que pol generador tenga polinomio primitivo como factor \rightarrow Siempre lo detectamos

Códigos con buenas propiedades detectoras y correctoras

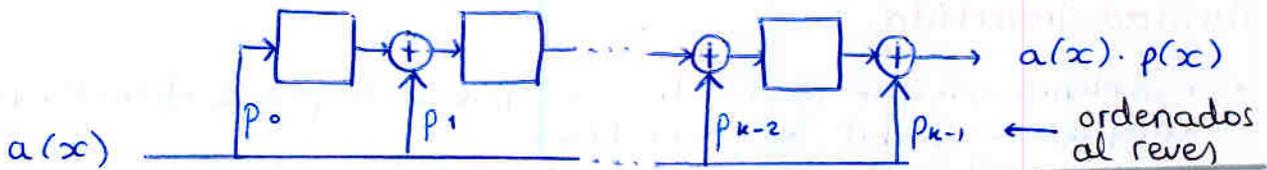
- Códigos BCH
- Códigos Reed-Solomon. generalización de BCH para códigos N-arios
 \rightarrow Se utiliza en los CD's (del orden de 100% redundancia)

3.5 Codificadores y Decodificadores cíclicos

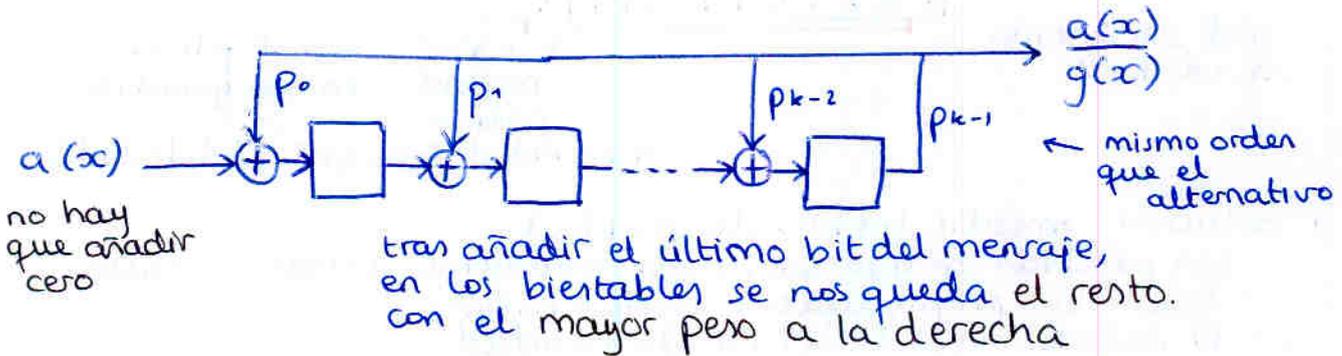
Multiplicadores polinómicos



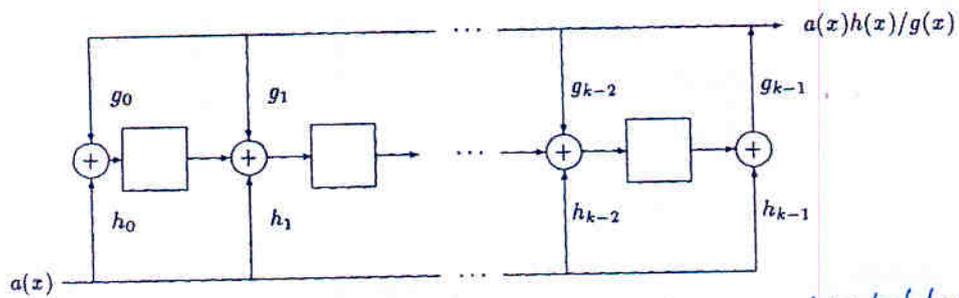
Multiplicador alternativo:



Divisor polinómico



Multiplicador/Divisor polinómico combinando los dos anteriores



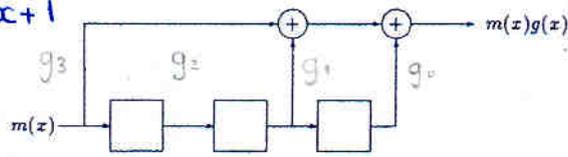
además saca el resto de la división en los biestables

utilidad: codificación códigos cíclicos sistemáticos

$$c(x) = m(x) \cdot x^{n-k} + \underbrace{\text{resto}\left(\frac{x^{n-k} \cdot m(x)}{g(x)}\right)}_{\text{CRC}}$$

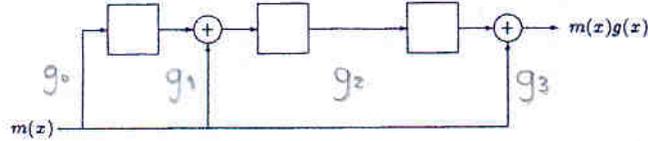
Codificador de la versión no sistemática del código cíclico

ej. $g(x) = x^3 + x + 1$



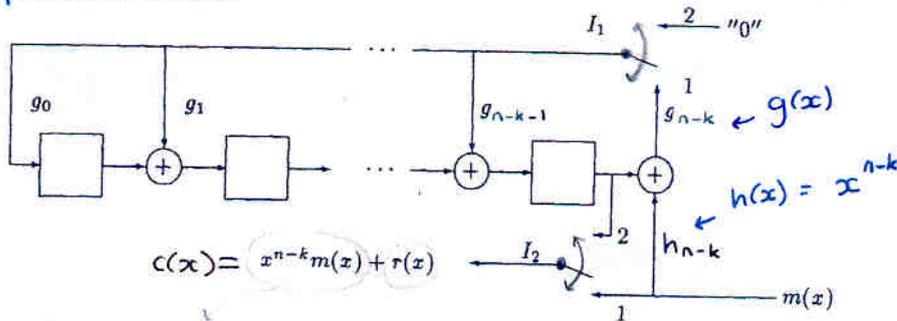
$$c(x) = m(x)g(x)$$

o con el multiplicador alternativo



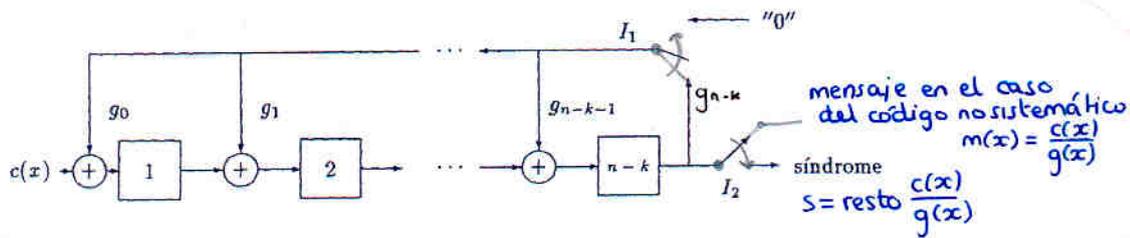
Codificador de la versión sistemática del código cíclico

multiplicador / divisor



- Posición 1: Introducir bits del mensaje \rightarrow van pasando directo a la salida al acabar tenemos el resto en los flip flops
- Posición 2: vamos extrayendo los bits del resto (de los flip-flops)

Decodificador (tanto para versión sistemática como no sistemática)



4. Códigos convolucionales

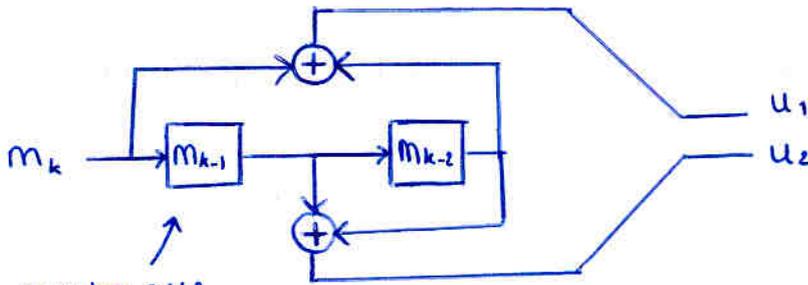
- Es un código lineal NO BLOQUE.
- Cada palabra código (lo que tx el transmisor) se fabrica para CADA mensaje, utilizando los K últimos mensajes (el mensaje actual y los $K-1$ anteriores)

CUIDADO:
No confundir con un código bloque que agrupe K mensajes, ya que eso no sacaría una palabra por cada mensaje

ejemplo $C(n, k, K) = C(2, 1, 3)$

(suelen ser 2, 1)

$m \xrightarrow[n]{1} \text{---} \xrightarrow[k]{2} u$
 $u_1 = m_k \oplus m_{k-2}$
 $u_2 = m_{k-1} \oplus m_{k-2}$



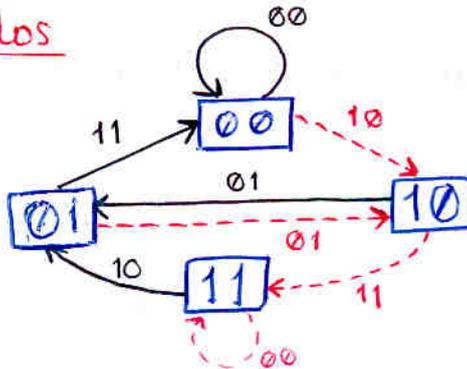
registro que almacena un mensaje. En éste caso al ser $k=1$ (como casi siempre en la práctica) no es otra cosa que un flip flop

Los valores iniciales de los biestables (determinarán el 'estado inicial') deberá conocerlos el receptor.

4.2 Principales Representaciones

• Diagrama de Estados

- entrada 0
- entrada 1



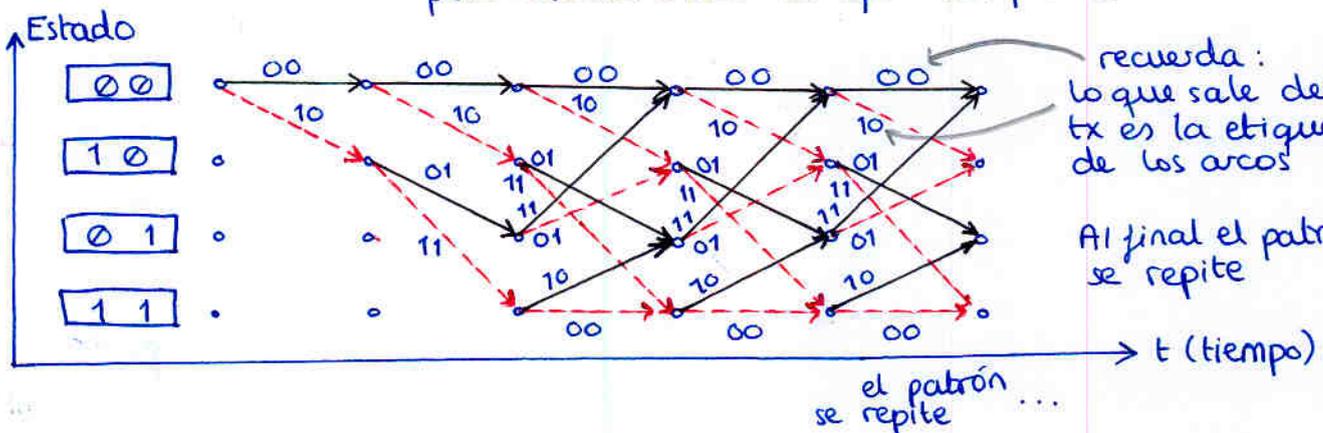
cada estado se nombra según el contenido de los $K-1$ flip flops

• de cada estado salen 2^k arcos

• la etiqueta del arco indica la palabra código saliente

• Rejilla de Trellis

EXACTAMENTE igual al diagrama de estados pero añadiendo el eje temporal



recuerda: lo que sale del tx es la etiqueta de los arcos

Al final el patrón se repite

Al acabar todos los mensajes, el tx habrá enviado una secuencia de palabras código llamada rama

Se dice que una rama es 'legal' si ésta concuerda con el código (i.e. aparece en la rejilla). Si no, se llama ilegal.

El código convolucional es lineal; suma de ramas legales siempre resultan en una rama legal

4.3 Decodificación

Los códigos convolucionales están pensados para SIEMPRE corregir

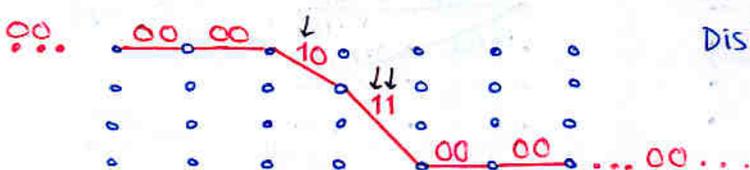
El receptor recibe una rama y le pasa al nivel superior los mensajes correspondientes a la rama legal más cercana a la recibida (máxima verosimilitud)

i.e. la que, suponiendo que ha sido enviada por el tx, haya tenido que sufrir menos bits erróneos para convertirse en la rama recibida.

Distancia libre (nombre que recibe la distancia mínima)

Viene dada por el peso ^{nº de unos} de la rama que menor peso tenga (i.e. menor número de unos txdos) sin contar la rama 00...000...

Ya que las ramas dependen de la longitud total del mensaje que se quiera enviar, podrían potencialmente ser infinitas, por tanto, la rama de menor peso es la que 'sale' un momento de la secuencia 00...00 y vuelve a ella cuanto antes

en el ej. anterior  Distancia libre: 3

Decodificación de máxima verosimilitud

El receptor pasa al nivel superior la rama legal más cercana a la recibida.

Generalmente hay que esperar a recibir toda la rama entera para poder decantarse

- Fuerza bruta

Al acabar la rx, el receptor calcula para cada una de las ramas existentes/posibles (cantidad exponencial con el número de bits recibidos) los errores que se habrían producido suponiendo que habiéndose enviado dicha rama, ésta se haya convertido en la rama recibida.

Pasa al nivel superior la rama con menor cantidad de errores.

Este método minimiza la probabilidad de error, pero es inviable computacionalmente.

• Decodificación de Viterbi

El resultado es el mismo que por fuerza bruta (minimiza probabilidad de error) pero es más 'ligero' computacionalmente

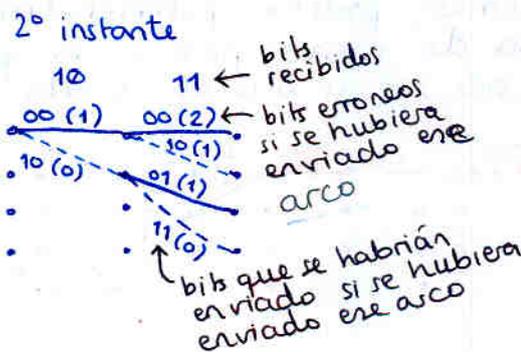
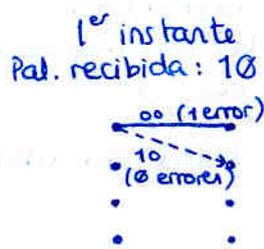
En cada instante de tiempo, en los estados donde haya más de una forma de llegar a ellos, deshecha el camino menos probable (no todo el camino, sólo la parte que sea exclusiva para llegar al estado bajo estudio o que solo tenga arcos salientes que hayan sido tachados previamente)

El camino menos probable se obtiene comparando los bits que se enviarían si el receptor fuese por ese camino con los bits que se han recibido: El menos probable es el que más bits erróneos tenga.

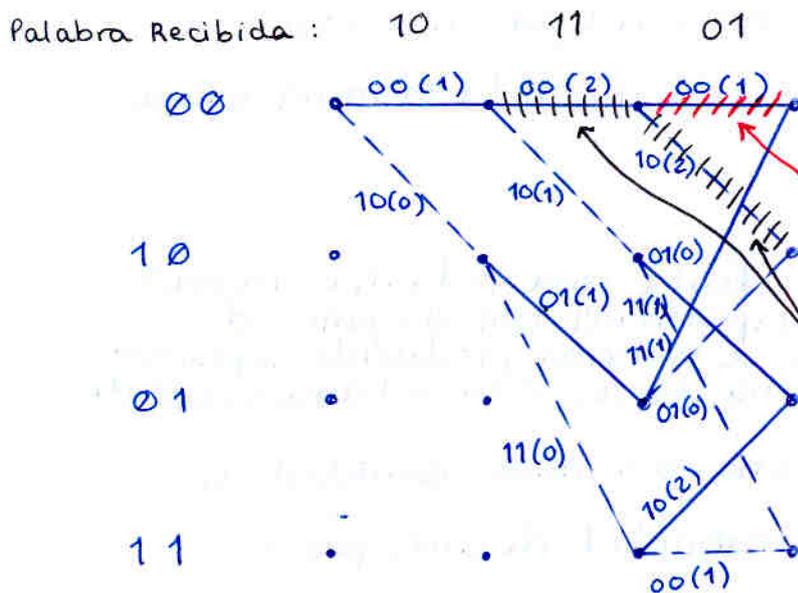
Si dos caminos son equiprobables, ¡elegimos uno al azar!
En los códigos convolucionales nunca nos conformamos con detectar errores, sólo con corregir.

ejemplo: recibimos la rama 10 11 01 11

Vamos instante por instante comprobando los estados a los que se podría llegar por más de un camino (hasta el tercer instante no ocurre esto)
Además ponemos en cada arco el número de errores de ese arco



3^o instante



Para el primer estado hay dos formas de llegar a el

00 00 00 → 1+2+1 errores = 4 errores

10 01 11 → 0+1+1 errores = 2 errores

→ tachamos primer camino (sólo hasta el 2^o instante)

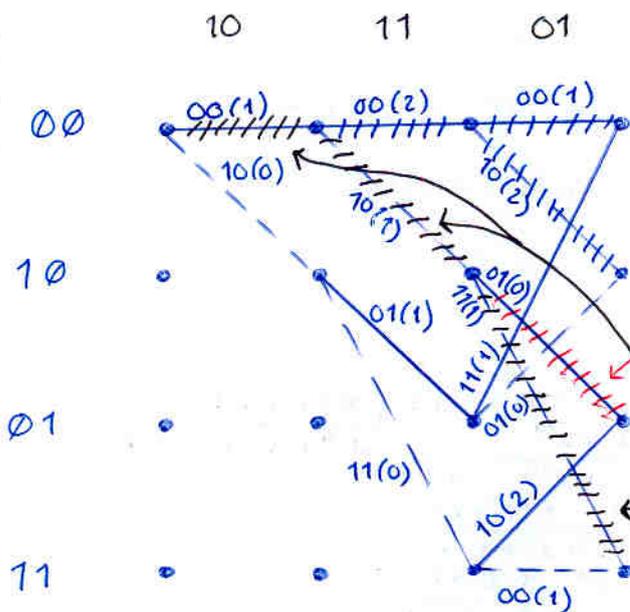
Segundo estado (10)

caminos:

00 00 10 (5 errores)

10 01 11 (2 errores)

→ tachamos 1^o camino, que combinado con lo q ya habia tachado, nos permite tachar mas.



- Tercer estado (01)
caminos:
00 10 01 = 2 errores
10 11 10 = 2 errores

Tachamos uno al azar!

- Cuarto Estado (11)
caminos:

00 10 11 = 3 errores
10 11 00 = 1 errores

Tachamos el 1º camino, que junto a todo lo tachado anteriormente nos permite tachar bastante

Ahora ya podríamos pasar al siguiente instante, y de nuevo tachar el camino menos probable para cada estado

Al acabar toda la recepción entera, tendríamos cuatro caminos restantes, escogeríamos el de máxima verosimilitud.

En cualquier instante en el algoritmo de Viterbi (excluyendo el transitorio inicial) tenemos tantas secuencias de ramas supervivientes como número de estados ($= 2^{\text{num registros} \cdot \text{bits por registro}}$)

Esto se debe a que siempre vamos estado por estado escogiendo la secuencia más probable para llegar a él.

En general no podemos decidir el valor de ningún bit hasta que no lleguemos al final de la transmisión (donde escogemos la trama más probable de entre las que quedan, que serán una por cada estado) pero se dan casos en los que sí que se pueden ir decidiendo los primeros bits conforme avanzamos en el algoritmo.

El algoritmo de Viterbi es óptimo en cuanto a mínima probabilidad de error y en cuanto a espacio de almacenamiento requerido para ejecutarlo.

Problemas

Sea un código cíclico $C(7, 4)$ con $g(x) = x^3 + x + 1$

X a) No existe un código de estas características
condiciones para ser $g(x)$ válido

- grado $n - k = 3$
- $C_{n-k} = G_0 = 1$
- resto $(\frac{x^n + 1}{g(x)}) = 0$

$$\begin{array}{r} x^7 + x^5 + x^4 + 1 \quad | \quad x^3 + x + 1 \\ \underline{x^7 + x^5 + x^4} \quad + 1 \\ x^5 + x^4 + 1 \\ \underline{x^5 + x^3 + x^2} \\ x^4 + x^3 + x^2 + 1 \\ \underline{x^4 + x^2 + x} \\ x^3 + x + 1 \\ \underline{x^3 + x + 1} \\ 0 // \end{array}$$

X b) 1111101 es palabra código

$$\begin{array}{r} x^6 + x^5 + x^4 + x^3 + x^2 + 1 \quad | \quad x^3 + x + 1 \\ \underline{x^6 + x^4 + x^3} \\ x^5 + x^2 + 1 \\ \underline{x^5 + x^3 + x^2} \\ x^3 + 1 \\ \underline{x^3 + x + 1} \\ x // \end{array}$$

resto $(\frac{r(x)}{g(x)}) \neq 0 \Rightarrow r(x)$ no es palabra código

1111101 no es palabra código, y utilizando la tabla estandar para corregir errores esa palabra estara asociada a un error

- ✓ c) simple
- X d) doble

i.e. ¿El síndrome de la palabra recibida corresponde al síndrome (i.e. está en la misma fila) de un patrón de error simple corregible o a uno doble corregible?

Para calcular el síndrome s : $g(x) \rightarrow G_{ns} \rightarrow G_s \rightarrow H^T \rightarrow s = r \cdot H^T$

$$g(x) = x^3 + x + 1 = 1011$$

recuerda: $k \times n$

$$G_{ns} = \begin{pmatrix} 1011000 \\ 0101100 \\ 0010110 \\ 0001011 \end{pmatrix} \xrightarrow{\oplus} \begin{pmatrix} 1000 & 101 \\ 0100 & 111 \\ 0010 & 110 \\ 0010 & 011 \end{pmatrix} = G_s$$

matriz generadora sistemática

$$H^T = \begin{pmatrix} P \\ I \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Con la matriz H^T podemos obtener MUCHA información del código.

RECUERDA: d_{\min} = mínimo número de filas que hay que sumar de H^T para obtener el vector nulo

- no hay fila $\vec{0}$ $\rightarrow d_{\min} \neq 1$
- no hay dos filas iguales $\rightarrow d_{\min} \neq 2$
- necesitamos mínimo tres filas para obtener el vector $\vec{0}$ sumándolas

$\hookrightarrow d_{\min} = 3 \Rightarrow$ detecta todos los patrones de error de 1 o 2 bits

Recuerda:

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor = 1 \Rightarrow \text{Corrige todos los patrones de error de un bit}$$

(no quiere decir que no corrija todos los patrones de error de dos bits!)
Podría ser que corrigiera algunos. (eso sí; no todos)

Síndrome de $r = 1111101$

$$s = r \cdot H^T$$

$$s = r \cdot H^T = (1111101) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{matrix} \text{sumar todas} \\ \text{las filas} \\ \text{(elemento a)} \\ \text{elemento} \\ \text{menos la} \\ \text{penúltima} \end{matrix} = (010)$$

El patrón de error asociado a una palabra no código de síndrome s es el patrón de error con ese mismo síndrome s .

(El patrón de error será el vector más probable (menos unos) que dé lugar al síndrome s)

En nuestro caso $r = 1111101$ está asociado al patrón de error $e = 0000010$ (ambos tienen síndrome 010)

De hecho: todos los posibles síndromes ($001, 010, 011, 100, 101, 110, 111$) están asociados a patrones de error simples. (H^T contiene todas las filas posibles)

\Rightarrow En este código todos los errores de 1 bit son corregibles

Ni siquiera existen los patrones de error dobles. Un error doble en el canal llevaría al receptor a interpretarlo como error simple.

\downarrow
llamados
códigos perfectos

Código cíclico $g(x) = x^3 + x + 1$ con $n=7$
Detecta y corrige todos los errores simples

2. En un código binario convolucional $C(2,1,3)$

Cuidado: a veces se llama palabras código a las posibles ramas (la salida ante un nuevo bit de mensaje calculada a partir del actual y los $k-1$ anteriores)

Otras veces (más correctamente según el profesor) se considera palabra código a una secuencia de ramas válida.

- a) Si las palabras código son $(00), (01), (10), (11)$:
 - ✓ No todas las secuencias posibles de ellas son válidas, de hecho en cada estado sólo hay dos ramas válidas salientes.
 - Precisamente en eso se basa.
- b) No existe pues $k > n \rightarrow$ FALSO. Recuerda siempre $C(n, k, K)$
- c) La suma de dos secuencias de palabras código es una secuencia de palabras código.
 - ✓ Si, ya que los códigos convolucionales que hemos estudiado son lineales (se podrían hacer no lineales generando las ramas de una manera no lineal a partir de los k registros. ej: inversor)
- d) Las palabras código se calculan a partir de k dígitos binarios.
 - Falso: se calculan a partir de K dígitos binarios

3. Un buen código bloque lineal será aquel que:

consiga una distancia mínima grande con pocos bits de redundancia.

4. Código $C(7,4)$ con matriz generadora G es código cíclico?

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

En principio vemos que G no tiene la forma de G_{ns} , pero podrían ser distintas versiones de un mismo código (mismo subespacio de palabras código generado)

Para ello podemos suponer que $g(x)$ sea el que viene dado por la última fila de G . $g(x) = 1011 = x^3 + x + 1$ y generar con él la matriz G_{ns} y sistematizarla a G_s para ver si coincide con la matriz G sistematizada

Suponemos $g(x) = x^3 + x + 1 \rightarrow G_{ns} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \xrightarrow{\text{Sistematización}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = G_s$

es igual a la calculada a partir de $G \rightarrow$ el código es el mismo!

De hecho podemos pasar de G_{ns} a G utilizando combinaciones lineales válidas

$$G_{ns} = X \cdot G$$

↑
matriz invertible

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \xrightarrow{\text{Combinaciones}} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = G_{ns}$$

$$X = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

invertible
(filas y columnas independ.)

TEMA 4. Modulaciones Digitales

1 Introducción

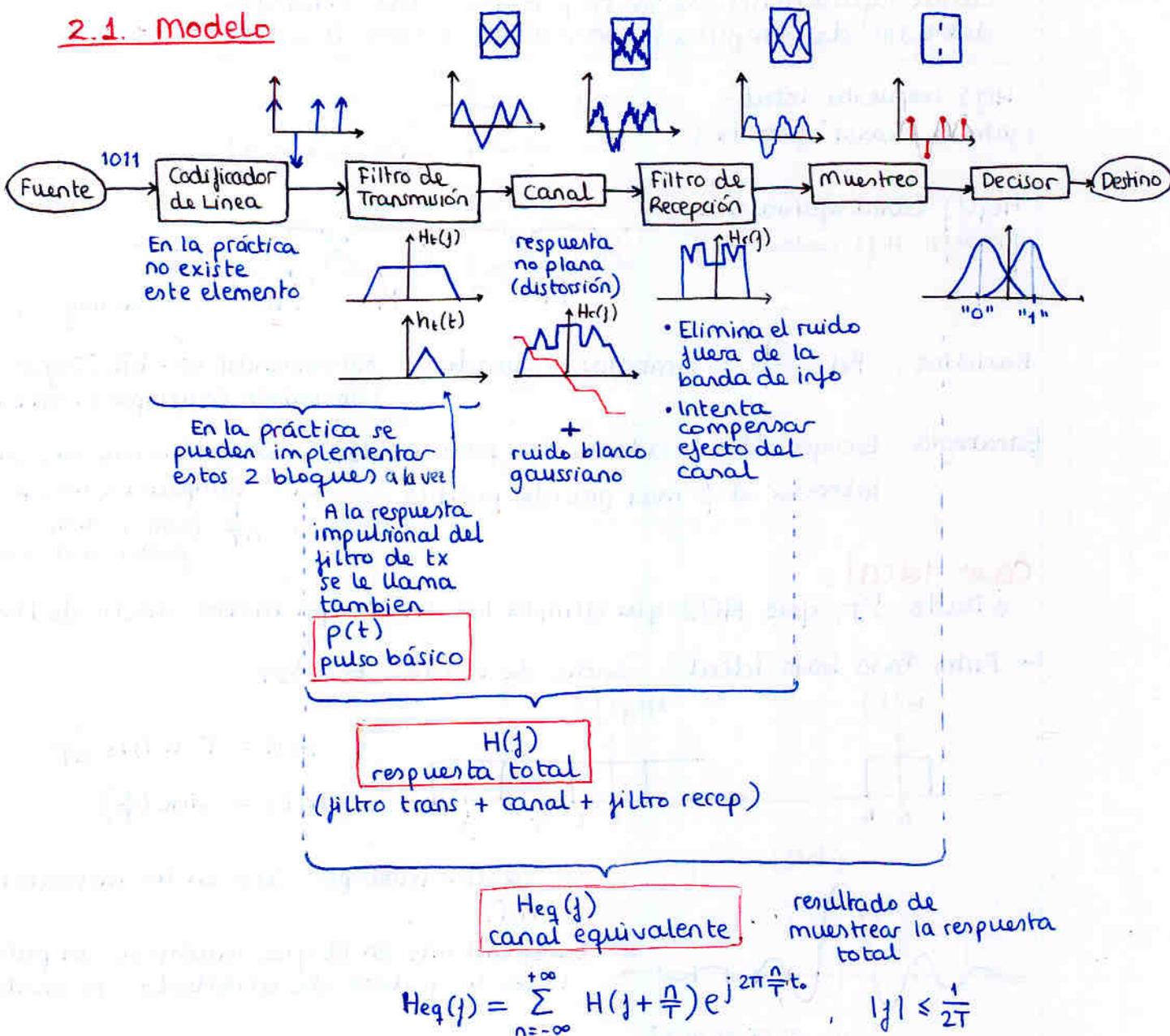
No confundirse: Modulación Digital → la señal es continua, pero los parámetros que la definen toman valores discretos.

Tipos

- Modulación Digital Banda Base: pulsos diferentes para símbolos diferentes.
 - Modulación Digital paso-banda: parámetros de senoide (A, ω, φ) diferentes para símbolos diferentes.
- ej: ASK, PSK, FSK, ...
 (equivalentes a AM, PM, FM pero se usa SK para denotar modul. digit)
 shift keying

2 Modulación digital banda base

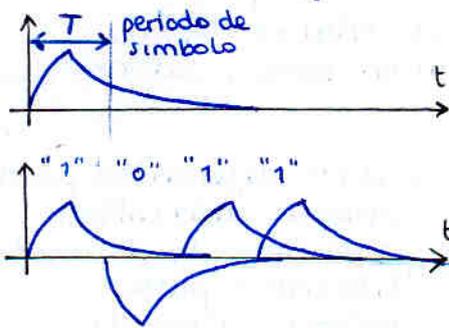
2.1. Modelo



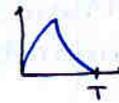
2.2 Interferencia entre símbolos y criterios de Nyquist

IES: Interferencia Entre Símbolos
ISI: Inter-Symbol Interference

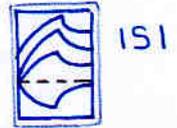
$h(t)$ total:
filtro tx
+ canal
+ filtro rx



¿Porque no utilizar que acaben en su propio periodo?



Requiere más ancho de banda.
Caso ideal no ISI: pulso cuadrado
↳ ancho de banda infinito

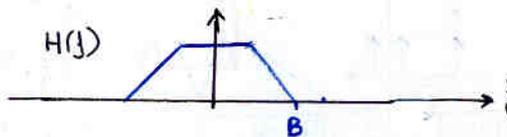


Hay menos margen para el ruido

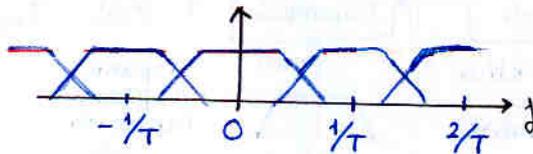
1er Criterio de Nyquist

Para eliminar la IES:
Canal equivalente de la respuesta total ($H_{eq}(f)$)
debe ser de amplitud constante, y fase lineal

$H(f)$ respuesta total
(filtro tx + canal + filtro rx)



$H_{eq}(f)$ canal equivalente
(repetir $H(f)$ cada $1/T$)



Baudios: $Bd = \frac{1}{T}$: símbolos/segundo

NO confundir con bits/segundo
(un símbolo no siempre es un bit)

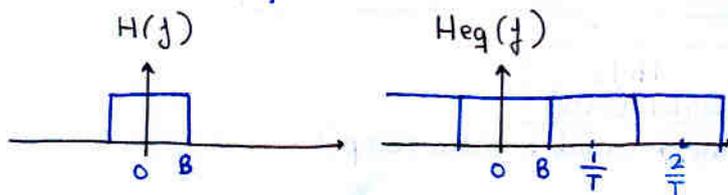
Estrategia: Escoger $Bd = \frac{1}{T}$ adecuado para no ISI.
Interesa el $\frac{1}{T}$ más grande posible

NOTA: Si no hay ISI para $\frac{1}{T}$
tampoco la hay para
 $\frac{1}{nT}$ (como si elimináramos
pulsos en el caso $1/T$)

Caso Ideal:

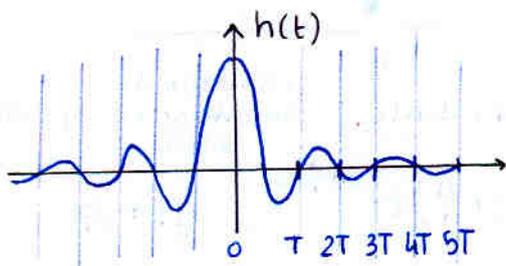
¿Dado $1/T$, que $H(f)$ que cumpla NO ISI ocupa menor ancho de banda?

↳ Filtro Paso bajo Ideal: ancho de banda $B = \frac{1}{2T}$



$$H(f) = T \text{ si } |f| \leq \frac{1}{2T}$$

$$h(t) = \text{sinc}\left(\frac{t}{T}\right)$$



La sinc para por cero en los instantes nT .

En el instante en el cual muestreo un pulso,
todos los pulsos de alrededor se anulan

2.3 Pulsos de Nyquist

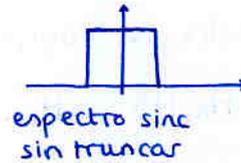
Problemas con pulsos sinc(t)

- Duración infinita
- No causal

⇒
Chapuzas

- Truncamos la respuesta
- Retardamos

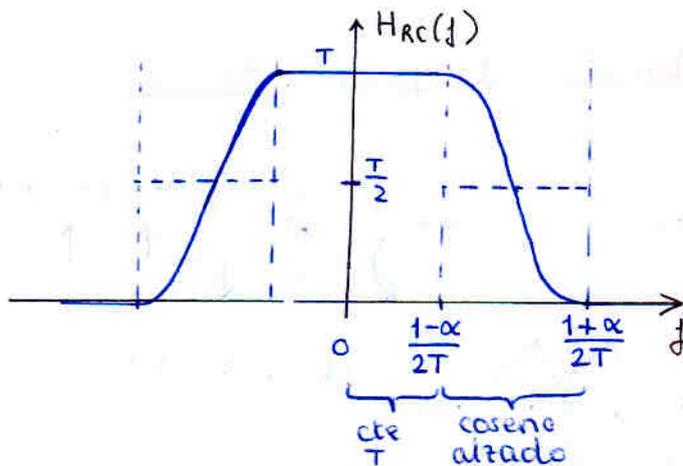
Al truncar la respuesta, quitamos mucha energía de las colas recortadas,
nuevo problema: aumenta el ancho de banda



Alivio del problema:

Empleamos sinc con las colas atenuadas (pulsos de Nyquist) que, aunque sin truncar requieren más ancho de banda que la sinc sin truncar, al truncarlos la energía que quitamos es mucho menor que cuando truncamos la sinc y su ancho de banda aumenta muy poco.

Pulsos de Nyquist o pulsos de coseno alzado

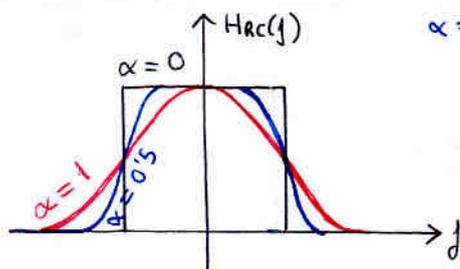


$$H(f) = \begin{cases} T & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\ \frac{T}{2} \left[1 + \cos \left(\frac{\pi}{2\alpha} (2|f|T + \alpha - 1) \right) \right] & \frac{1-\alpha}{2T} \leq |f| \leq \frac{1+\alpha}{2T} \\ 0 & \text{resto} \end{cases}$$

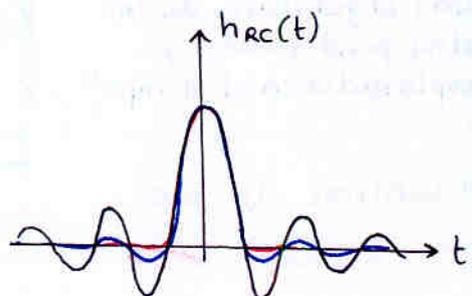
$$\Downarrow \text{TF}^{-1}$$

$$h(t) = \text{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\alpha\pi t}{T}\right)}{1 - \frac{4\alpha^2 t^2}{T^2}}$$

Parámetro $\alpha \in [0, 1]$



$\alpha=0 \Rightarrow \text{sinc}$



al aumentar α , el ancho de banda aumenta, pero al truncar la señal, el incremento de ancho de banda será menor cuanto mayor sea α

↳ Hay que llegar a un compromiso

Pulsos raíz cuadrada de Nyquist

Filtro adaptado a un pulso: el mejor filtro posible para que la relación (S/N) a su salida sea máxima cuando a la entrada tenemos el pulso en cuestión.

Se consigue haciendo que la respuesta impulsional del filtro coincida con la del pulso.

Pulso raíz cuadrada de Nyquist: $\sqrt{H_{rc}(f)}$

A diferencia de $H_{rc}(f)$, $\sqrt{H_{rc}(f)}$ si que tiene ISI (su canal equiv. no es plano) por eso no es adecuado usarlo como respuesta total;

Pero si hacemos:

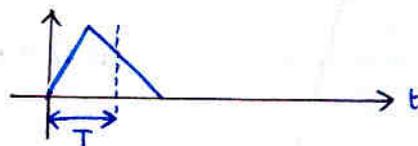


Matamos 2 pájaros de un tiro

- eliminamos ISI (filtro total = $H(f)$ sin ISI)
- minimizamos el ruido (el filtro de recepción está adaptado al pulso)

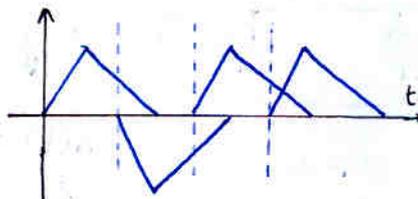
2.4 Señales banda base y diagrama de ojos

Pulso básico $p(t)$
(respuesta al impulso del
filtro de transmisión)



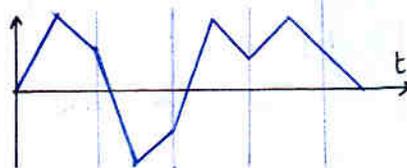
$$\sum_k a_k \delta(t - kT)$$

Pulsos ponderados
y desplazados en
el tiempo



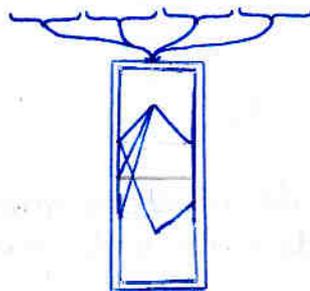
$$a_k p(t - kT)$$

señal en banda base:
"suma algebraica de los
pulsos ponderados y
desplazados en el tiempo"



$$s(t) = \sum_k a_k p(t - kT)$$

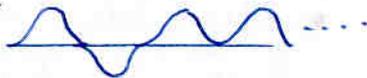
Diagrama de ojos:



2.5. Estudio espectral de la señal

Señales definidas en energía: energía finita
ej: pulso $p(t)$ 

↳ Permiten calcular la densidad espectral de energía.

Señales definidas en potencia: energía infinita
potencia finita
ej: modulación banda base
 $\sum_k a_k p(t-kT)$ 

↳ Permiten calcular la densidad espectral de potencia

Densidad Espectral de energía de un pulso

$$p(t) \xrightarrow{TF} P(j) \longrightarrow \boxed{E_p(j) = |P(j)|^2} \text{ J/Hz}$$

↓ cálculo de la energía en el tiempo

$$E = \int_{-\infty}^{\infty} |p(t)|^2 dt$$

↓ cálculo de la energía mediante el espectro

$$E = \int_{-\infty}^{\infty} E_p(j) df$$

} Teorema de Parseval

Densidad espectral de potencia de una señal banda base

$$s(t) = \sum_k a_k p(t-kT)$$

d.e.p. $\boxed{S(j) = |P(j)|^2 \frac{1}{T} \sum_{n=-\infty}^{\infty} R_a(n) e^{-j2\pi n T f}} \text{ W/Hz}$

NOTA: que no nos preocupen las exp complejas. Debes saber que $S(j)$ siempre es real, lo que ocurre es que $R_a(n)$ siempre es par y por tanto $e^{j2\pi n T f}$ siempre se suma con su conjugado $e^{-j2\pi n T f}$ resultando un número real

$R_a(n)$: autocorrelación de los a_k . Depende del código

$$\boxed{R_a(n) = E_{(k)} \{a_k \cdot a_{k-n}\}}$$

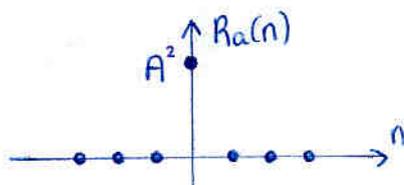
ejemplo: Código Linea polar

$n=0$

$$R_a(0) = E \{a_k \cdot a_k\} = E \{a_k^2\}$$

suponiendo símbolos equiprobables

$$= \frac{1}{2}(A^2) + \frac{1}{2}(-A)^2 = A^2$$



$n=1$

$$R_a(1) = E \{a_k \cdot a_{k-1}\}$$

si suponemos bits estadísticamente independientes

$$= E \{a_k\} \cdot E \{a_{k-1}\}$$

$$= [\frac{1}{2}(A) + \frac{1}{2}(-A)] \cdot [\frac{1}{2}(A) + \frac{1}{2}(-A)] = 0$$

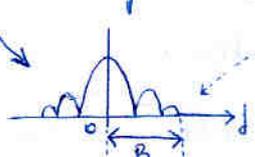
Deducimos: $R_a(n > 1) = 0$

Resumen $\left\{ \begin{array}{l} E_{p(t)}(j) = |P(j)|^2 \\ S(j) = |P(j)|^2 \cdot \underbrace{\frac{1}{T} \sum_n R_a(n) e^{-j2\pi n T f}}_{\text{única diferencia}} \end{array} \right.$

Ancho de banda

Intervalo de frecuencias positivas de una densidad espectral (energía o potencia) donde se cumple un criterio

si hablamos del pulso si hablamos de la señal en banda base



se anula fuera de la banda
primer nulo
cierto % de potencia/energía
-3 dB

Eficiencia espectral

$$E_f = \frac{R_M}{B} \left[\frac{\text{Bd}}{\text{Hz}} \right] = \left[\frac{\text{simb/s}}{\text{Hz}} \right]$$

ejemplo: pulsos coseno alzado con $\alpha=0 \Rightarrow \text{sinc}$

$$E_f = \frac{R_M}{B} = \frac{1/T_{\text{bit}}}{1/2T_{\text{bit}}} = 2 \text{ Bd/Hz}$$

$$E_f' = \frac{R_I}{B} = \frac{R_M \log_2 M}{B} \left[\frac{\text{bit/s}}{\text{Hz}} \right] \quad \text{a veces se expresa en bit/s}$$

$$E_f' = \log_2 M \cdot E_f$$

2.6. Señalización multinivel. $M > 2$

R_M : velocidad de modulación
 R_I : régimen binario

$$R_I = R_M \cdot \log_2 M$$

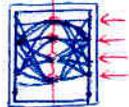
$$\left[\frac{\text{bit/s}}{\text{Bd}} \right] = \left[\frac{\text{simb/s}}{\text{Bd}} \right] \cdot \left[\frac{\text{bit/simb}}{\text{Bd}} \right]$$

una señalización M-aria debe transmitir menor número de símbolos/s que una señalización binaria para transmitir los mismos b/s

$M \uparrow \Rightarrow$ menor R_M (simb/s) para lograr mismo R_I (bit/s) \Rightarrow menor variación de la señal \Rightarrow menor ancho de banda

Recuerda: cuanto más rápido varía una señal más ancho de banda usa

$M \uparrow$
 ventaja \Rightarrow menor ancho de banda
 problema \Rightarrow más sensible al ruido (menos margen para equivocarse)



Energía media por símbolo

$$E_s = \frac{E_0 + E_1 + \dots + E_{M-1}}{M}$$

siendo $E_i =$ energía de símbolo i -ésimo $= \int_{-\infty}^{\infty} a_i^2 p(t)^2 dt$

$$E_s = \frac{M \cdot E\{a_k^2\} \cdot E_p}{M}$$

$E_i = a_i^2 \cdot E_p$ energía del pulso básico $p(t)$

energía media por símbolo (J) energía del pulso básico (J)

$$E_s = E_p \cdot E\{a_k^2\} \left[\frac{\text{J}}{\text{simb}} \right]$$

Energía media por bit

$$E_b = \frac{E_s}{\log_2 M} \left[\frac{\text{J}}{\text{bit}} \right]$$

2.7. Señalización por respuesta parcial

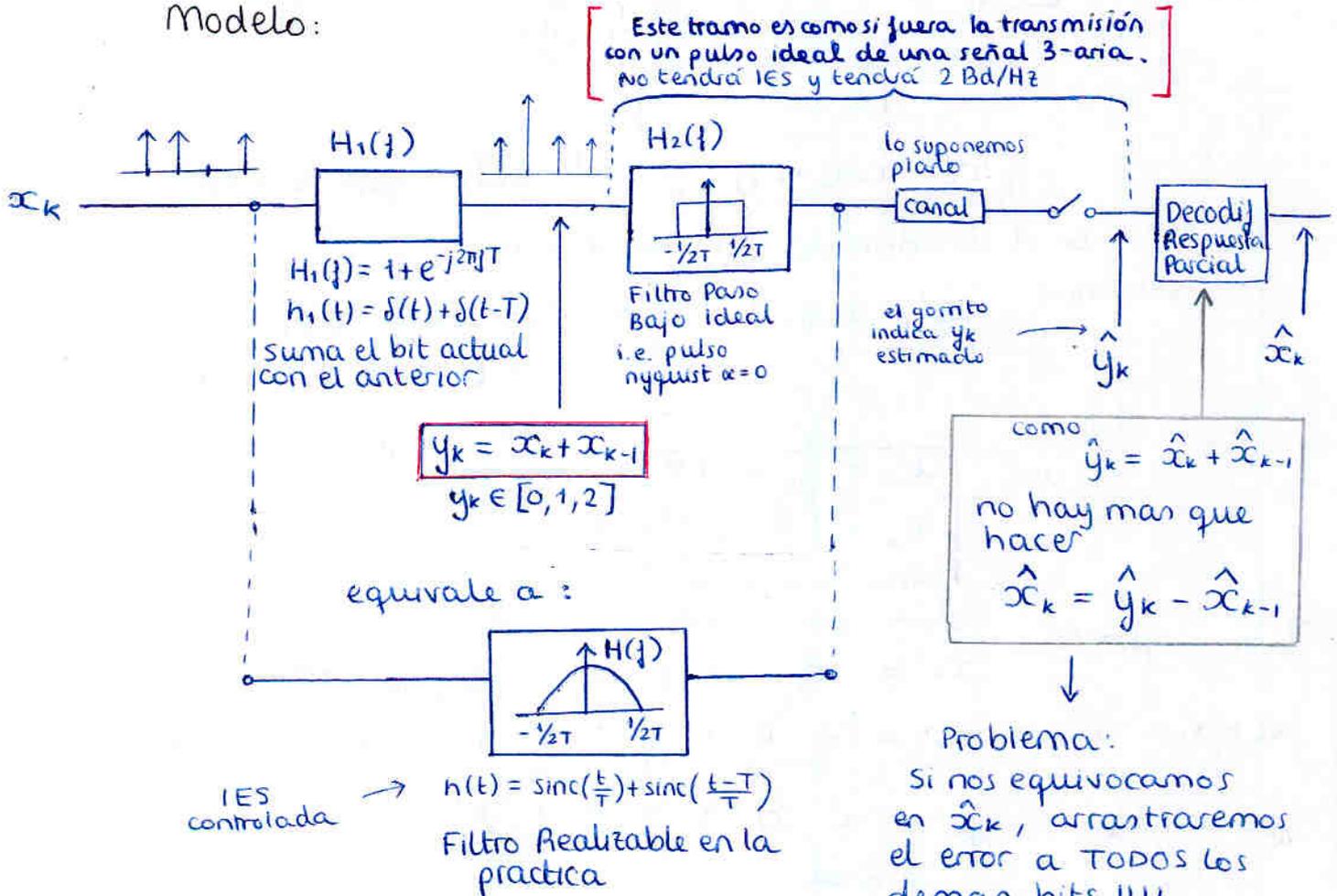
La máxima eficiencia espectral posible se consigue con pulso Nyquist $\alpha = 0$ y vale 2 Bd/Hz

→ ¿Podemos conseguirla con una respuesta total suave, más fácil de abordar?

Si que podemos, pero el canal equivalente ya no será plano y habrá IES, aunque será IES conocida y por lo tanto como si no la hubiera

Desventaja: incrementamos número de niveles en el receptor

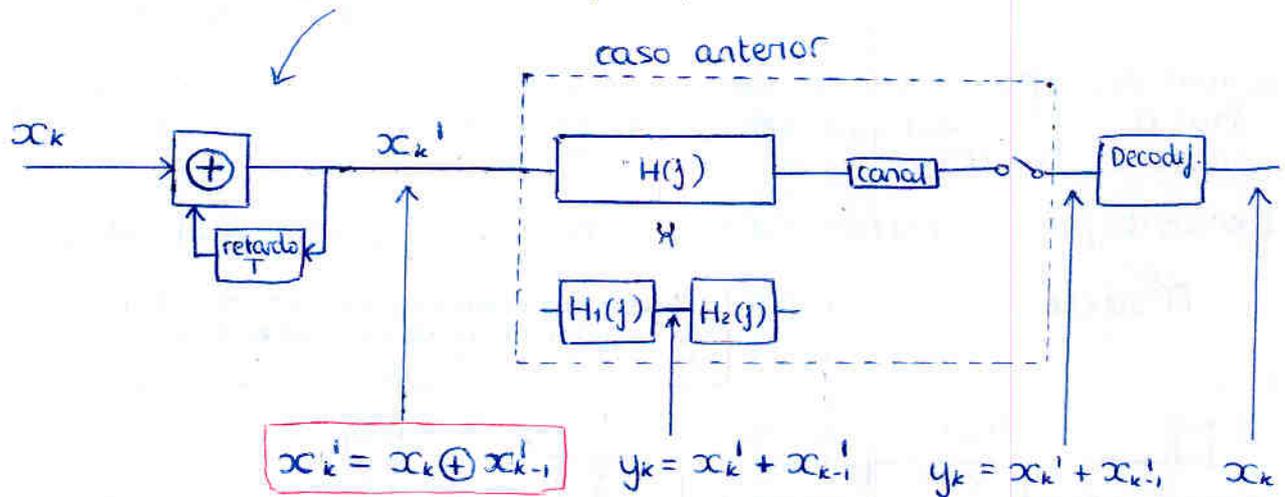
Modelo:



Precodificación

Soluciona totalmente el problema de que si nos equivocamos en un bit arrastramos el error

No hay mas que añadir al principio lo siguiente



En el decodificador esta vez, se hace:

Sabiendo

$$\begin{aligned} x_k' = x_k \oplus x_{k-1}' &\Rightarrow x_k = x_k' \oplus x_{k-1}' \\ &= (x_k' + x_{k-1}') \bmod 2 \\ &= y_k \bmod 2 \end{aligned}$$

i.e.

$$x_k = y_k \bmod 2$$

$$x_k = \begin{cases} 0 & y_k = 0, 2 \\ 1 & y_k = 1 \end{cases}$$

ejemplo:

$$\begin{aligned} x_k &= 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \\ x_k' = x_k \oplus x_{k-1}' &\Rightarrow x_k' = 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \\ y_k = x_k' + x_{k-1}' &\Rightarrow y_k = 0 \ 0 \ 1 \ 2 \ 1 \ 1 \ 2 \\ &\quad \left. \begin{array}{c} \\ \end{array} \right\} \text{canal} \\ y_k &= 0 \ 0 \ 1 \ 2 \ 1 \ 1 \ 2 \\ x_k = \begin{cases} 0 & y_k = 0, 2 \\ 1 & y_k = 1 \end{cases} &\Rightarrow x_k = 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \end{aligned}$$

precodificación
primer elemento es
arbitrario y conocido

coincide con el
mensaje original
y además no
arrastra errores

Codificación correlativa
o por respuesta parcial

$$y_k = x_k + x_{k-1} \Rightarrow$$

se la llama
duobinario o
respuesta parcial clase I

Existen otras
combinaciones

$$y_k = x_k + x_{k-2} \Rightarrow$$

duobinario modificado
clase IV

2.8 Ecuación

Anteriormente hemos estado buscando cual debería ser nuestro filtro total (transmisión + canal + recepción).

De forma que nuestro canal deseado

$$H(f)_{\text{deseado}} = H_T(f) \cdot \hat{H}_C(f) \cdot H_R(f)$$

Estimación de la respuesta del canal

Filtro fijo de compromiso, diseñado para el caso en que $H_C(f) = \hat{H}_C(f)$

Problema: En la práctica no conocemos la respuesta del canal; es más: ésta puede ser variable con el tiempo (pensar por ej en un móvil)

En la práctica

$$H(f)_{\text{deseada}} = H_T(f) \cdot H_C(f) \cdot H_R(f) \cdot H_E(f)$$

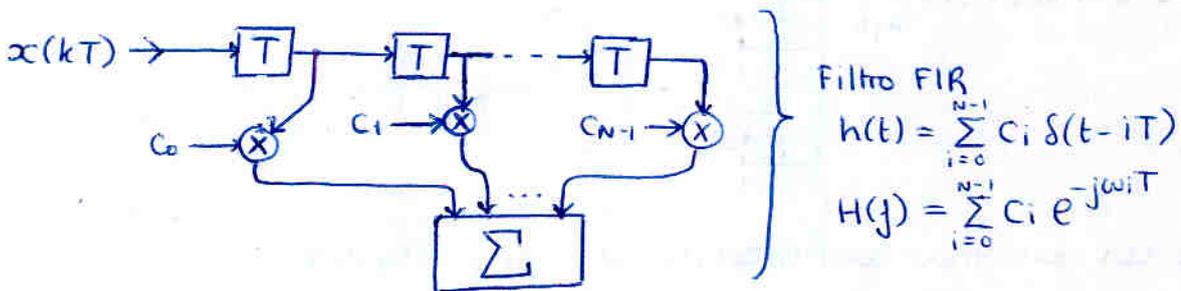
Filtro fijo de compromiso. Ecuación. Trata de compensar las variaciones del canal respecto del estimado
 (intuitivamente: $H_E(f) = \frac{\hat{H}_C(f)}{H_C(f)}$)

$H_R'(f)$
Filtro de Recepción adaptable

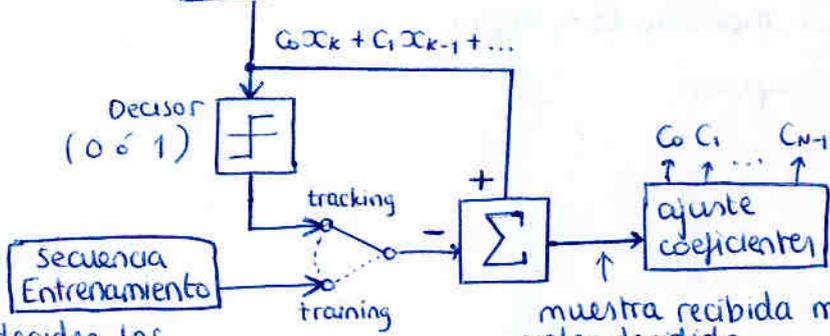
Tipos de ecualizadores $H_E(f)$

- Fijos (de compromiso) (i.e. $H_R(f)$ sin $H_E(f)$)
- Manuales (ajustables)
- Automáticos (se ajustan al principio con secuencia de entrenamiento)
- Adaptativos: (se ajustan al inicio y durante la comunicación)

Ecuación adaptativos



Filtro FIR
 $h(t) = \sum_{i=0}^{N-1} c_i \delta(t-iT)$
 $H(f) = \sum_{i=0}^{N-1} c_i e^{-j\omega iT}$

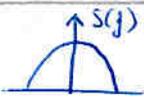
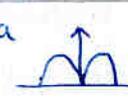


inicialmente se deciden los coeficientes por comparación muestra esperada / recibida

muestra recibida menos valor decidido
 ej: recibo 0'872 y supongo 1
 la diferencia es 0'128
 con ella puedo ajustar coeficientes

2.9 Códigos de línea

→ Adecuar el espectro (con $R_a(n)$) $S(f) = \frac{|P(f)|^2}{T} \sum_n R_a(n) e^{-j2\pi f n T}$

ejemplo, tenemos  pero nos gustaria  ya que el canal se porta mal en continua

→ Información de sincronismo

es importante en código en el que no ocurren cosas como ... 010000000 →  ?

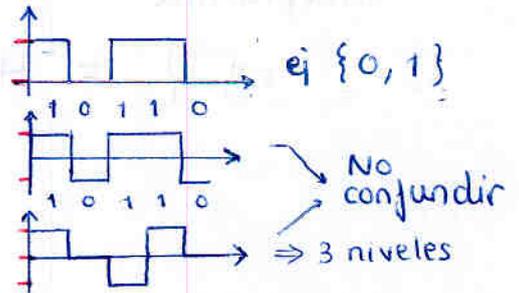
→ Control de errores ej: transiciones no permitidas

Clasificación

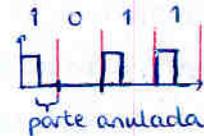
Uni polar : Todos los pesos del mismo signo

Polar : Pesos simétricos ej $\{-1, 1\}$

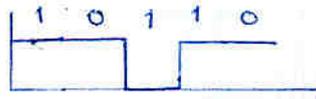
Bipolar : El 1 va cambiando su peso
 (AMI) alternate mark inversion
 0 → 0
 1 → ±1
 ↳ logra anular la continua



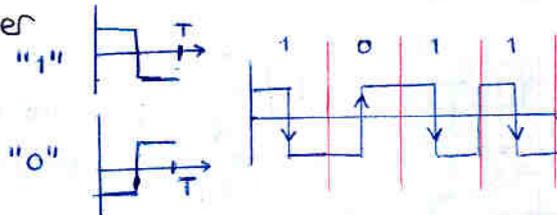
RZ (return zero) : anulamos un porcentaje del periodo
 NRZ (non return zero)



Diferencial M (mark) } el -1 alterna el ~~0~~ anterior, y el 0 lo repite
 S (space) }
 al revés el 0 alterna y el 1 repite



Bifase L o Manchester



Se pueden combinar distintas de estas propiedades

ej: Polar Manchester Diferencial

ej: RZ bipolar

Codigos Pseudoternarios

utilizan 3 niveles para transmitir binario

↳ Alfabéticos

$(mB - nT)$ m bits Binarios \rightarrow n deltas Ternarias

(tb pueden ser M-arias)
i.e. $mB - nM$

ej: 2B2T
Tabla alfabeto

00	\rightarrow	- +
01	\rightarrow	0+ ó 0-
10	\rightarrow	+0 ó -0
11	\rightarrow	+ -

} eligiendo una u otra para minimizar la continua

Pueden requerir incremento de velocidad ej 4B-3T

↳ No alfabéticos

\rightarrow BnZS: Bipolar with n-Zeros Substitution

sustituyen una secuencia larga de ceros por un determinado código que suele incluir violaciones de polaridad para distinguirse

implica retardo en codificación de n bits

ej: B8ZS: 0000 0000 \rightarrow 000V B0VB

B: bipolar: respetar alternancia
V: violación de polaridad, tienen misma polaridad que ultimo pulso

ej: B3ZS: 000 \rightarrow B0V ó 00V la que deje numero impar de pulsos B entre dos V

\rightarrow HDBn: High Density Bipolar with n zeros max

es igual que el BnZS pero CUIDADO con la notación, esta permite HASTA n zeros: i.e. sustituye n+1 zeros

ej: HDBn 000...00 \rightarrow B0...0V ó 00...0V
q deje nº impar de B's entre 2 V's

Entrelazados

Multiplexación entre 2 o más codigos de linea

Densidad espectral de potencia

de algunos códigos de línea (utilizando pulsos cuadrados)

Recuerda

pulso:

$$p(t) = \text{rect}(t/T) \rightarrow |P(f)|^2 = T^2 \text{sinc}^2(fT) \text{ J/Hz}$$

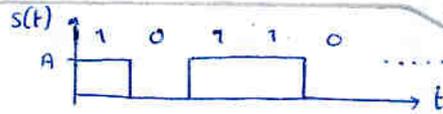
señal
banda base

$$s(t) = \sum a_k p(t-kT)$$

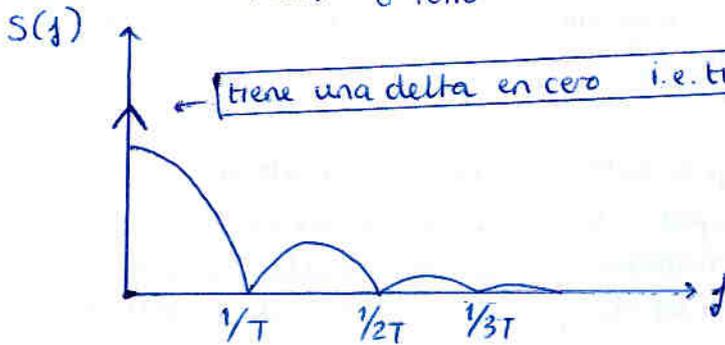
$$\hookrightarrow S(f) = \frac{|P(f)|^2}{T} \sum_{n=-\infty}^{\infty} R_a(n) e^{-j2\pi f n T} \text{ W/Hz}$$

el código determinará esto

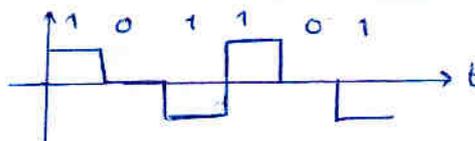
ej: NRZ unipolar



$$R_a(n) = \begin{cases} A^2 & n=0 \\ 0 & \text{resto} \end{cases}$$



ej NRZ bipolar



$$R_a(0) = \frac{1}{2} 0 + \frac{1}{2} A^2 = \frac{A^2}{2}$$

$$R_a(1) = R_a(-1) = \frac{1}{4} (0 \cdot 0) + \frac{1}{4} (0 \cdot (\pm A)) + \frac{1}{4} ((\pm A) \cdot 0) + \frac{1}{4} ((\pm A) \cdot (\mp A)) = 0 + 0 + 0 - \frac{1}{4} A^2 = -\frac{1}{4} A^2$$

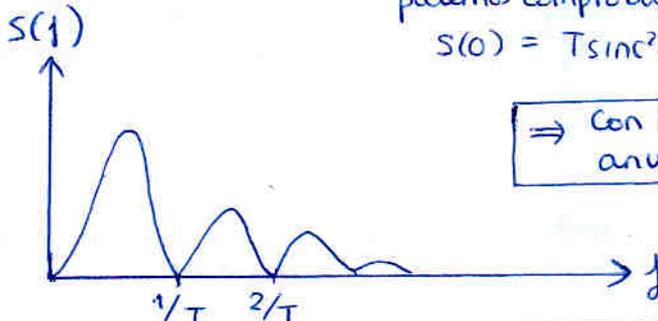
$$R_a(2) = 0$$

$$R_a(n) = \begin{cases} A^2/2 & n=0 \\ -A^2/4 & n=\pm 1 \\ 0 & \text{resto} \end{cases}$$

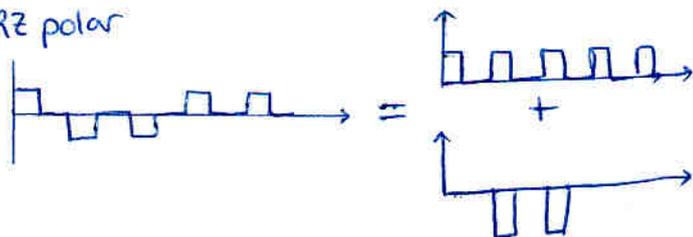
podemos comprobar como en $f=0$ se anula

$$S(0) = T \text{sinc}^2(0) \left[-\frac{A^2}{4} + \frac{A^2}{2} - \frac{A^2}{4} \right] = 0$$

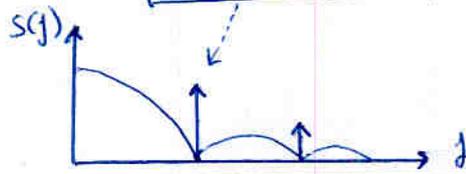
⇒ Con la NRZ bipolar logramos anular la continua



RZ polar



periodica ⇒ deltas que nos facilitarán el sincronismo!!!



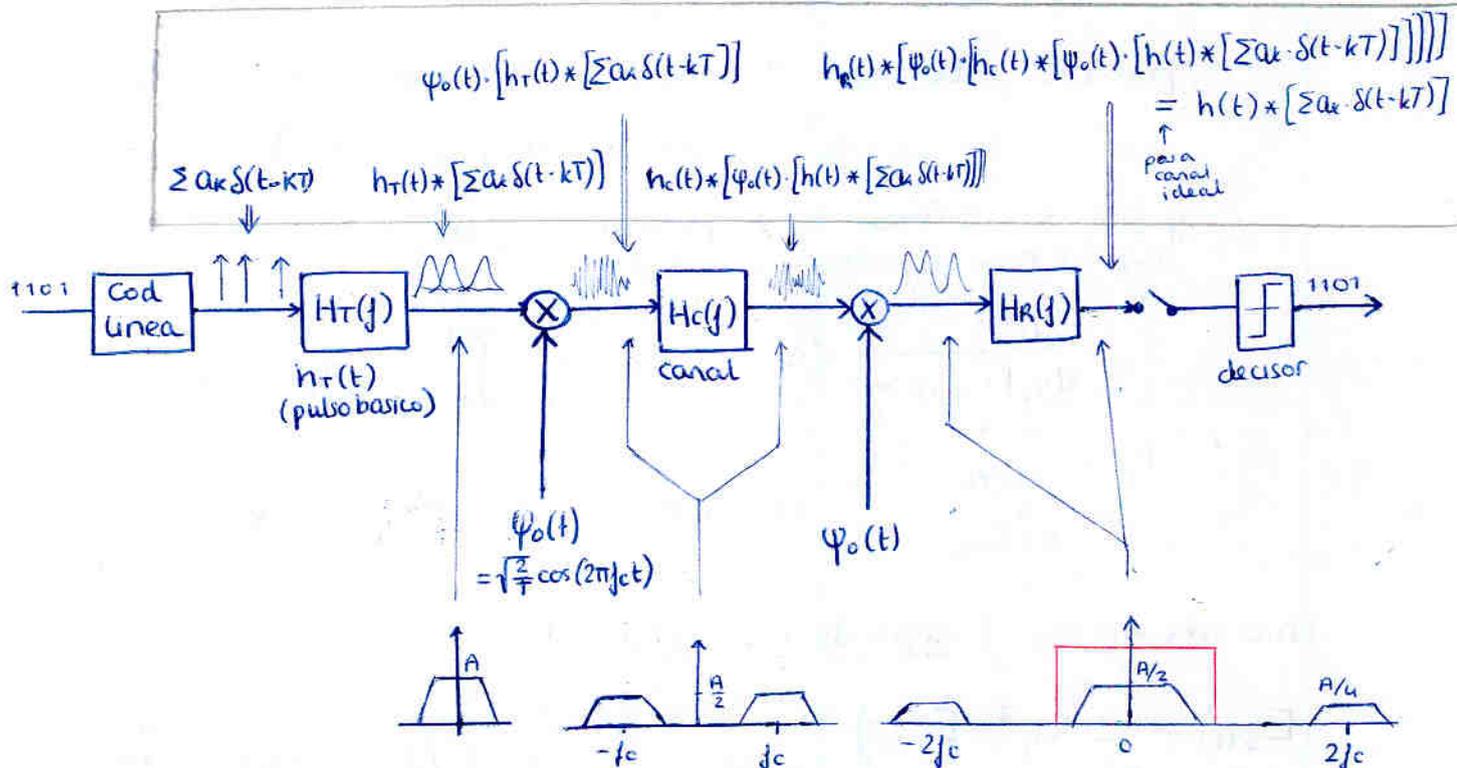
3. Modulación Digital Paso Banda

Hay varias formas de ver el proceso de modulación digital banda base.

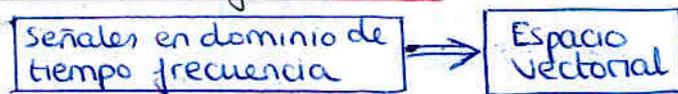
- Modificar algun(os) parámetros que definen a una portadora (sinusoide) acorde con los datos o señal banda base
 - ej: amplitud: ASK
 - fase: PSK
 - frecuencia: FSK
- Híbridos

- Modelo analítico paso banda:

Multiplicar la señal banda base por una portadora f_c antes y después del canal



- Caracterización geométrica



Para ello definimos:

Producto Escalar: $\langle \psi_i(t), \psi_j(t) \rangle = \int_0^T \psi_i(t) \cdot \psi_j(t) dt$

Y de esa definición podemos definir una 'base' de señales ortogonales (igual que haríamos con vectores) si se cumple:

$$\{\psi_0(t), \psi_1(t), \dots, \psi_{N-1}(t)\} \text{ es ortogonal} \iff \langle \psi_i(t), \psi_j(t) \rangle = \int_0^T \psi_i(t) \cdot \psi_j(t) dt = \begin{cases} 0 & i \neq j \\ \int_0^T \psi_i^2(t) dt = E_{\psi_i(t)} & i = j \end{cases}$$

$= k_j \cdot \delta_{ij}$
 \downarrow
 $E_{\psi_j(t)}$

\hookrightarrow delta de Kronecker $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Tenemos un conjunto de M señales $s_0(t), s_1(t), \dots, s_{M-1}(t)$ $M \geq N$ aplicando método de Gram-Schmidt N Funciones base ortogonales $\psi_0(t), \psi_1(t), \dots, \psi_{N-1}(t)$

Entonces todas las señales 's' se pueden expresar como combinación lineal de funciones ψ

$$s_0(t) = a_{00} \cdot \psi_0(t) + a_{01} \cdot \psi_1(t) + \dots + a_{0N-1} \cdot \psi_{N-1}(t)$$

$$s_1(t) = a_{10} \psi_0(t) + a_{11} \psi_1(t) + \dots + a_{1N-1} \cdot \psi_{N-1}(t)$$

⋮

es decir:

$$s_i(t) = \sum_{j=0}^{N-1} a_{ij} \cdot \psi_j(t)$$

podemos interpretar s_i como un vector usando la base ψ_j

$$\vec{s}_i = (a_{i0}, a_{i1}, a_{i2}, \dots, a_{iN-1})$$

y podemos calcular los coeficientes a_{ij} de forma sencilla (igual que si fueran vectores)

$$a_{ij} = \frac{1}{\langle \psi_j, \psi_j \rangle} \cdot \langle s_i, \psi_j \rangle = \frac{1}{k_j} \cdot \int_0^T s_i(t) \cdot \psi_j(t) dt$$

$$\begin{aligned} &= \frac{1}{k_j} \\ &= \frac{1}{\|\vec{\psi}_j\|^2} \end{aligned}$$

$$\langle \psi_j(t), \psi_j(t) \rangle = \int_0^T \psi_j^2(t) dt = E_{\psi_j(t)} = k_j$$

En consecuencia, Energía de una señal $s_i(t)$

$$E_{s_i(t)} = \sum_{j=0}^{N-1} (a_{ij}^2 \cdot E_{\psi_j(t)})$$

$$\begin{aligned} &= \sum_{j=0}^{N-1} a_{ij}^2 \cdot k_j \\ &= \sum_{j=0}^{N-1} a_{ij}^2 \cdot \|\vec{\psi}_j\|^2 \end{aligned}$$

Demonstración

$$\begin{aligned} E_{s_i(t)} &= \int_0^T s_i(t)^2 dt = \int_0^T \left[\sum_j a_{ij} \psi_j(t) \right]^2 dt = \left(\begin{array}{l} \text{multiplicar 2 sumatorios} \\ (a+b) \cdot (a+b) \\ = a \cdot a + a \cdot b + b \cdot a + b \cdot b \end{array} \right) \\ &= \int_0^T \left[\sum_j a_{ij} \psi_j(t) \right] \left[\sum_k a_{ik} \psi_k(t) \right] dt = \sum_j \sum_k (a_{ij} a_{ik} \int_0^T \psi_j(t) \cdot \psi_k(t) dt) \\ &= (\text{por ser } \psi \text{ una base ortogonal}) = \sum_j \sum_k a_{ij} a_{ik} \cdot (k_j \cdot \delta_{jk}) \\ &= \sum_j a_{ij}^2 \cdot k_j \end{aligned}$$

Si $E_{\psi_j(t)} = k_j = 1 \Rightarrow$ La base $\{\psi_0, \psi_1, \dots\}$ es ortonormal $\|\vec{\psi}_j\| = 1$

$$\begin{aligned} \text{base ortonormal} &\Rightarrow E_{s_i(t)} = \sum_j a_{ij}^2 = \|\vec{s}_i\|^2 \\ \|\vec{\psi}_j\| = 1 & \end{aligned}$$

La energía de la señal s es igual al cuadrado del módulo de su vector

Transformada de Fourier Generalizada

es como se denomina todo lo anterior

$$\int_0^T \psi_i(t) \cdot \psi_j(t) dt = k_j \delta_{ij} \quad \begin{matrix} k_j = \|\psi_j\|^2 \\ \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \end{matrix}$$

$$s_i(t) = \sum_{j=0}^{N-1} a_{ij} \psi_j(t) \quad \text{vector } \bar{s}_i = (a_{i0}, a_{i1}, \dots)$$

$$a_{ij} = \frac{1}{k_j} \int_0^T s_i(t) \psi_j(t) dt \quad \longrightarrow \quad E_{s_i(t)} = \sum_j a_{ij}^2 \cdot \|\psi_j\|^2$$

↑
coordenadas de s_i
en la base formada
por ψ_j

Base ortogonal y ortonormal

Base ortogonal $\rightarrow E_{s_i(t)} = \sum_j a_{ij}^2 \cdot \|\psi_j\|^2$

Base ortonormal $\rightarrow E_{s_i(t)} = \sum_j a_{ij}^2 = \|\bar{s}_i\|^2$
 \uparrow
 $\|\psi_j\| = 1$ mucho más sencillo

A partir de una base ortogonal es inmediato obtener una base ortonormal sin más que dividir cada vector de la base por su módulo o lo que es lo mismo, por la raíz cuadrada de su energía

$$\psi'_i(t) = \frac{\psi_i(t)}{\|\psi_i(t)\|} = \frac{\psi_i(t)}{\sqrt{E_{\psi_i(t)}}}$$

- no es más que un escalado.
- Es normalizar cada vector

ejemplo:

Base ortogonal $\psi_0(t) = \cos(2\pi f_c t)$
 $\psi_1(t) = \sin(2\pi f_c t)$

$$E_{\cos(2\pi f_c t)} = \int_0^T \cos^2(2\pi f_c t) dt = \frac{T + \frac{\sin(4\pi f_c T)}{4\pi f_c}}{2} = \frac{T}{2} (1 + \text{sinc}(4f_c T))$$

$$= \begin{cases} = \frac{T}{2} & \text{si } T = \frac{k}{4} \cdot T_c \quad \text{i.e. el periodo de simbolo } T \text{ es un multiplo de un cuarto del ciclo del coseno} \\ \simeq \frac{T}{2} & \text{si } 4f_c T \gg 1 \quad \text{i.e. el periodo de simbolo } T \text{ es mucho mas grande que un cuarto del ciclo del coseno} \end{cases}$$

Podemos obtener base ortonormal sin más que dividir cada vector de la base por la raíz cuadrada de la energía

Base ortonormal: $\psi'_0(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c T)$ $\left(\begin{array}{l} \text{siempre y cuando} \\ T = \frac{k}{4} \cdot T_c \\ \text{ó } T \gg \frac{1}{4} \cdot T_c \end{array} \right)$
 $\psi'_1(t) = \sqrt{\frac{2}{T}} \sin(2\pi f_c T)$

3.2. Técnicas básicas

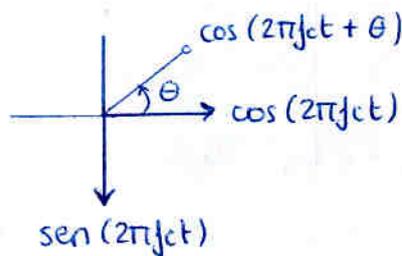
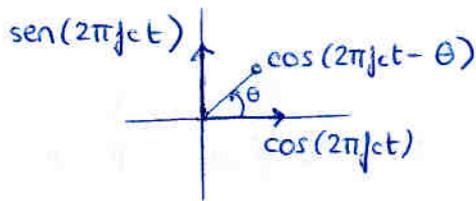
Aspectos geométricos de sinusoides, Misma frecuencia

Funciones base
base ortonormal

$$\begin{cases} \psi_0(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t) \\ \psi_1(t) = \sqrt{\frac{2}{T}} \sin(2\pi f_c t) \end{cases}$$

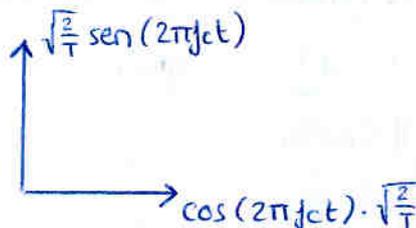
Cualquier senoide (de cualquier fase o amplitud) de frecuencia f_c se puede escribir como combinación de esta base.

Se suele representar de alguna de estas formas



Para recordarlo es útil saber / deducir que $\sin(2\pi f_c t) = \cos(2\pi f_c t - \frac{\pi}{2})$

La componente del coseno se llama: componente en fase
la del seno se llama: componente en cuadratura

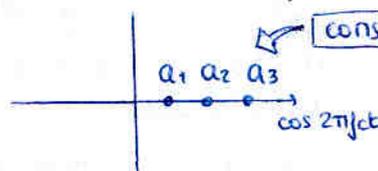


Pero cuidado; la ortogonalidad o no, depende del intervalo T en el que definimos el producto escalar (que en nuestro contexto es el periodo de simbolo)

$$\int_0^T \psi_0(t) \cdot \psi_1(t) = \frac{2}{T} \int_0^T \cos(2\pi f_c t) \cdot \sin(2\pi f_c t) = \sin(2\pi f_c t) \cdot \text{sinc}(2f_c T) = \begin{cases} 0 & \text{si } T = \frac{k}{2} \cdot T_c \\ & T_c = 1/f_c \\ \neq 0 & \text{si } T \gg \frac{1}{2} T_c \end{cases}$$

ejemplo modulación en este subespacio

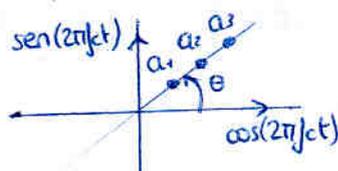
M-ASK unipolar



constelación de la modulación

$$\begin{aligned} a_1(t) &= A_1 \cdot \sqrt{\frac{2}{T}} \cos 2\pi f_c t \\ a_2(t) &= A_2 \cdot \sqrt{\frac{2}{T}} \cos 2\pi f_c t \\ &\vdots \end{aligned}$$

en realidad podrian tener cualquier fase (que sea la misma)



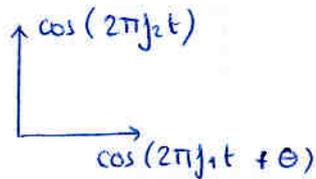
$$\begin{aligned} a_1(t) &= A_1 \cdot \sqrt{\frac{2}{T}} \cos(2\pi f_c t - \theta) \\ a_2(t) &= A_2 \cdot \sqrt{\frac{2}{T}} \cos(2\pi f_c t - \theta) \\ &\vdots \end{aligned}$$

Aspectos geométricos de sinusoides de distintas frecuencias

- Distinta fase y frecuencia:

Para que $\cos(2\pi f_1 t)$ y $\cos(2\pi f_2 t + \theta)$ sean ortogonales se debe cumplir:

$$\int_0^T \cos(2\pi f_1 t + \theta) \cdot \cos(2\pi f_2 t) dt \simeq 0 \text{ si } \begin{cases} f_1 - f_2 = k_1 \cdot \frac{1}{T} \\ f_1 + f_2 \gg 1 \end{cases} \text{ ambas condiciones simultáneamente}$$



condición de ortogonalidad que deben cumplir las frecuencias de 2 cosenos si estos no tienen la misma fase.

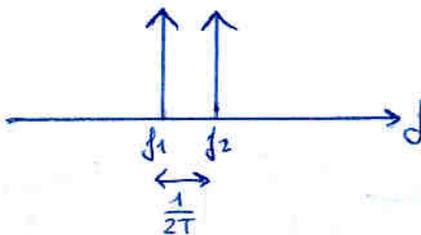
- Distinta frecuencia; misma fase:

Para que dos cosenos de frecuencias f_1 y f_2 que tengan la misma fase (i.e. empiecen a la vez) sean ortogonales debe cumplirse

$$\int_0^T \cos(2\pi f_1 t) \cos(2\pi f_2 t) dt = \frac{T}{2} [\text{sinc}(2(f_1 - f_2)T) + \text{sinc}(2(f_1 + f_2)T)]$$

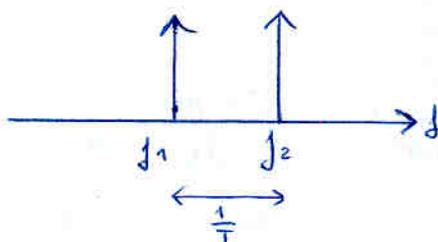
$$\begin{cases} = 0 \text{ si } \begin{cases} f_1 - f_2 = k_1 \cdot \frac{1}{2T} \\ f_1 + f_2 = k_2 \cdot \frac{1}{2T} \end{cases} \text{ en la teoría} \\ \simeq 0 \text{ si } \begin{cases} f_1 - f_2 = k_1 \cdot \frac{1}{2T} \\ f_1 + f_2 \gg 1 \end{cases} \text{ en la práctica} \end{cases}$$

Por lo tanto



$f_1 - f_2 = \frac{1}{2T}$ es la mínima separación que podemos tener entre tonos para que sean ortogonales.

Y sólo en el caso de que ambos tengan la misma fase



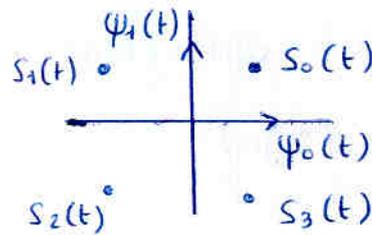
Si no podemos asegurar que la fase sea la misma (bastante habitual) entonces la separación entre frecuencias mínima pasa a ser el doble

$$f_1 - f_2 = \frac{1}{T}$$

Receptor coherente

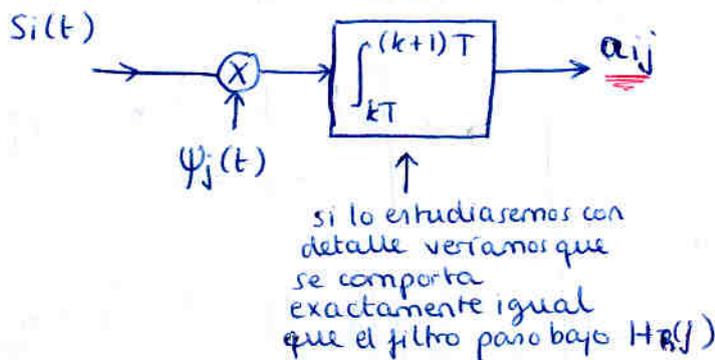
Los receptores coherentes necesitan una réplica de la portadora usada en el emisor (o más en general, de las funciones base utilizadas)

El receptor recibe una de entre M posibles señales (modulación M-aria) en el subespacio creado por la base ortogonal



Se basa en la idea de 'proyectar' u obtener la componente de la señal en cada eje para decidir que símbolo es.

sabiendo $a_{ij} = \frac{1}{k_j} \int_0^T s_i(t) \cdot \psi_j(t) dt$ basta implementar



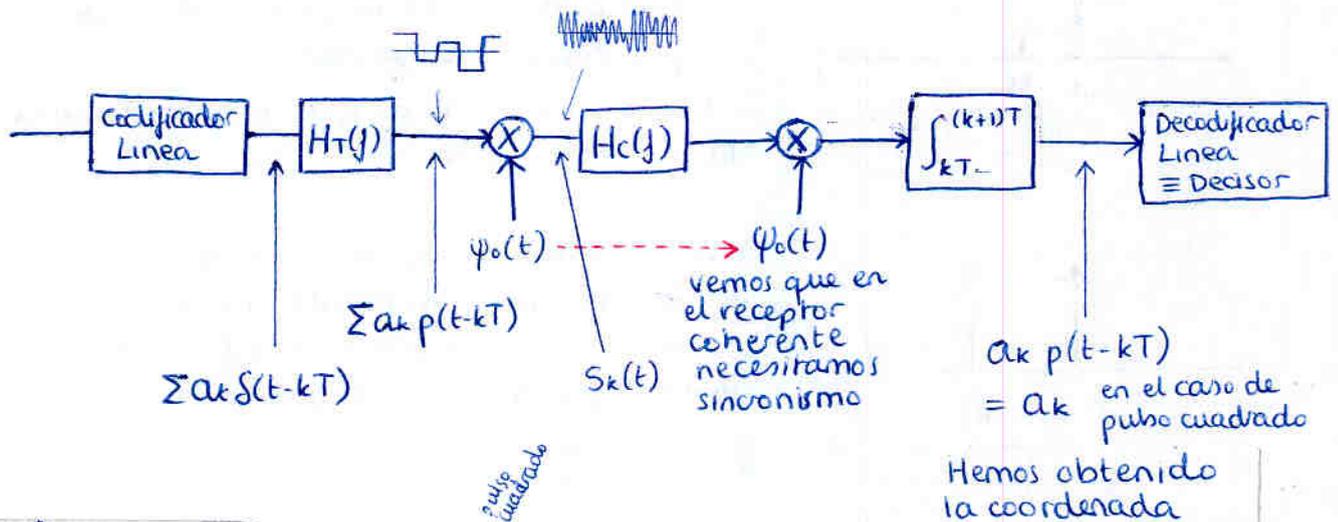
para obtener los coeficientes de s_i no habría más que repetir ésta rama $\forall j$

ejemplo: modulación unidimensional ASK

ASK polar:



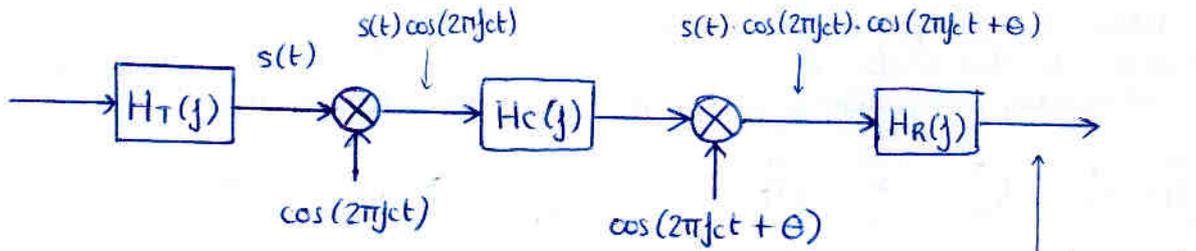
$$\psi_0(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t)$$



Dem:

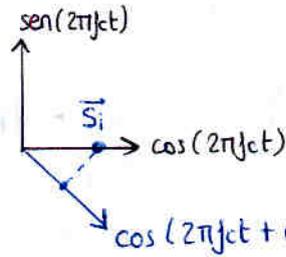
$$\int_{kT}^{(k+1)T} S(t) \cdot \psi_0(t) dt = \int_{kT}^{(k+1)T} \psi_0^2(t) \sum_K a_k \Pi\left(\frac{t-kT}{T}\right) dt = a_k \int_{kT}^{(k+1)T} \psi_0^2(t) dt = a_k$$

Desincronización de la fase en detector sincrónico



Intuitivamente lo que está pasando es que el receptor está utilizando como función sobre la que proyectar $\cos(2\pi jct + \theta)$

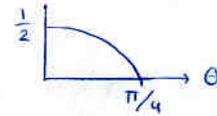
i.e.



referencia sobre la cual proyecta \vec{S}_i .

Se reduce el módulo del coeficiente extraído, lo cual afectará directamente a la (S/N)

suponiendo canal ideal y $H_R(j)$ F.P.B ideal se obtiene
 $= \frac{1}{2} \cos \theta s(t)$

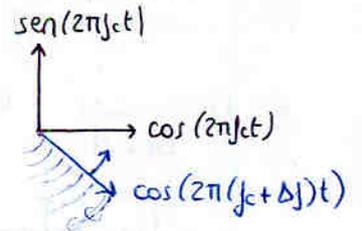


Desincronización de la frecuencia

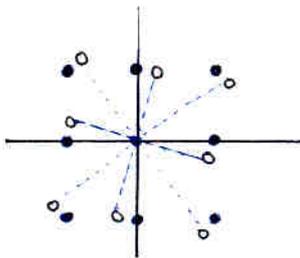
Si la frecuencia se desincroniza, entonces ya que $f(t) = \frac{1}{2\pi} \frac{d\psi(t)}{dt}$ lo que pasará es que la fase irá aumentando.

Es decir, la referencia del receptor irá girando!!

Por suerte la frecuencia es relativamente fácil de controlar, a diferencia de la fase



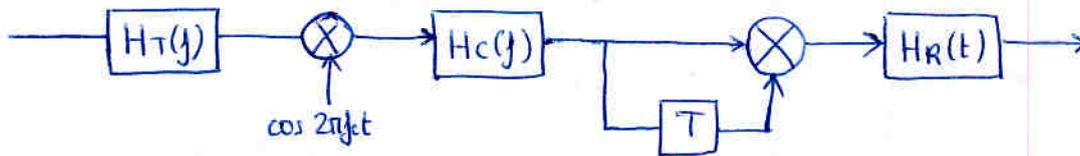
El efecto de una desincronización de la fase sobre la correlación es como si la rotara



Así, con un menor ruido podríamos llegar a confundir símbolos

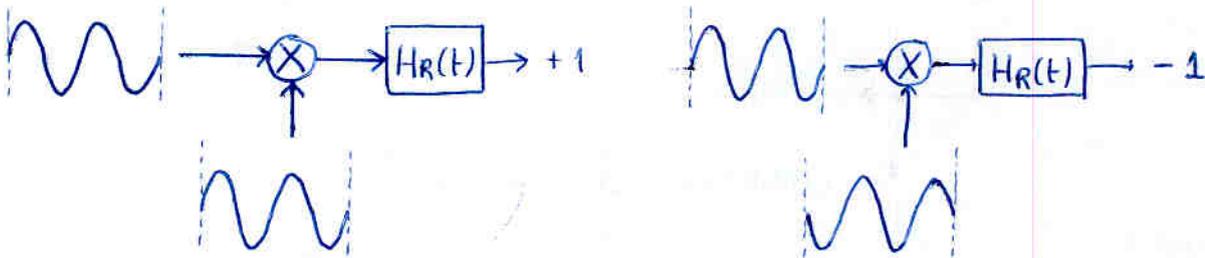
Receptor no coherente

- No necesitan una réplica de la portadora.
- La emulan utilizando el símbolo anterior recibido - ^{esquema diferencial}
Es más sencillo, pero funciona peor (por ej los símbolos tendrán ruido)



ejemplo: DPSK (PSK diferencial)

emisor {
 "0" → no cambia la fase actual
 "1" → cambia la fase actual

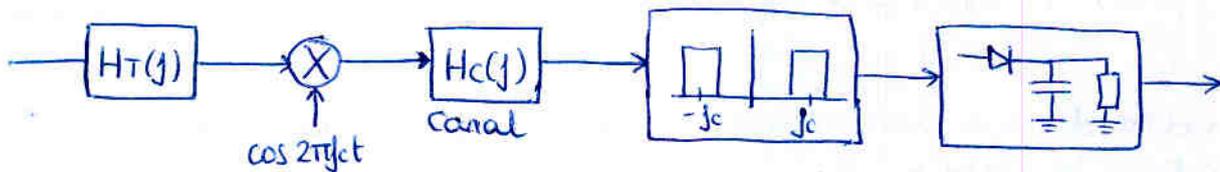


i.e. Si el símbolo recibido lo proyectamos sobre uno igual (i.e. símbolo anterior era igual al recibido) su proyección será +1

Si el símbolo recibido lo proyectamos sobre el de fase contraria (i.e. símbolo anterior era fase distinta) su proyección será -1

Hemos logrado decodificar el mensaje que tenía la fuente del emisor

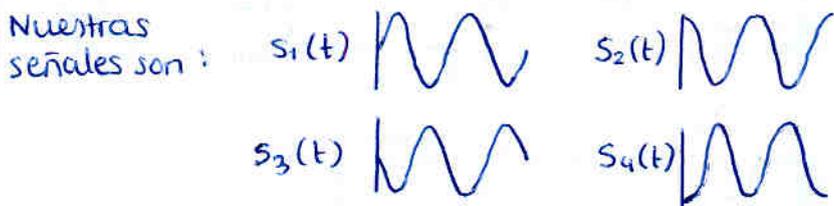
- Para una ASK unipolar también sirve un simple filtro paso banda + detector de envolvente



No sirve para ASK polar

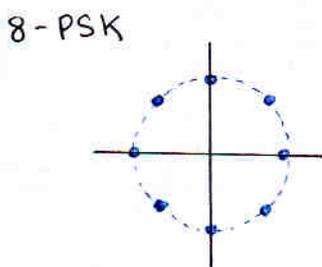
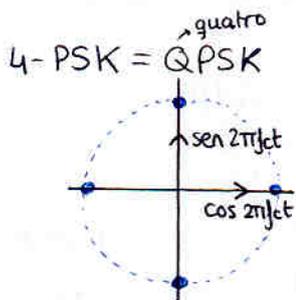
Modulaciones Bidimensionales

ejemplo 4-PSK
¿Porqué 2 dimensiones y no una?



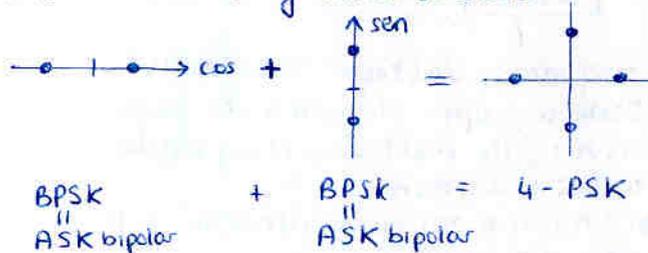
Al hacer Gram-Schmidt obtenemos dos señales: $\psi_0(t)$ y $\psi_1(t)$ p.ej sen y cos

Un espacio con 2 funciones base es un espacio bidimensional



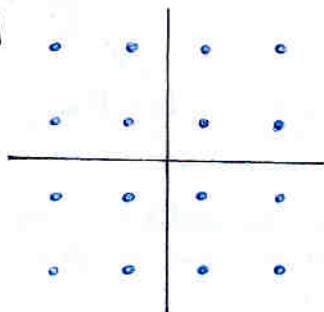
Démonos cuenta que una 4-PSK es como 2 BPSK's en cuadratura, una utilizando como portadora el coseno y otra el seno

En general una 2^n -PSK son n BPSK usando distintas fases como portadoras y sumándose

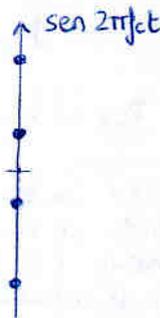


QAM (Quadrature Amplitude Modulation)

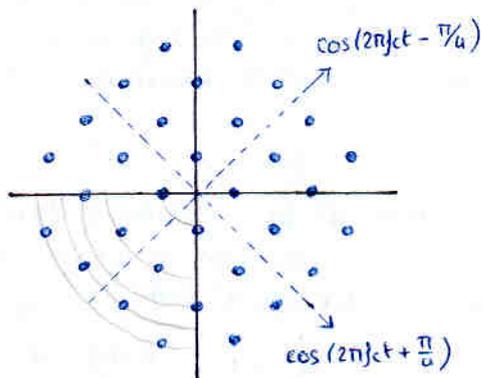
16-QAM



No es mas que dos 4-ASK en cuadratura! (De ahí el nombre)



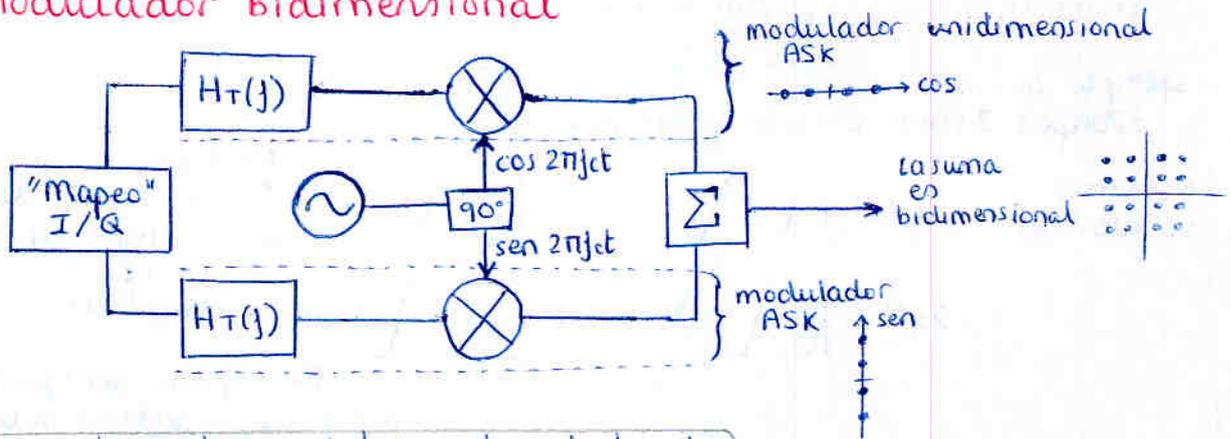
32-QAM



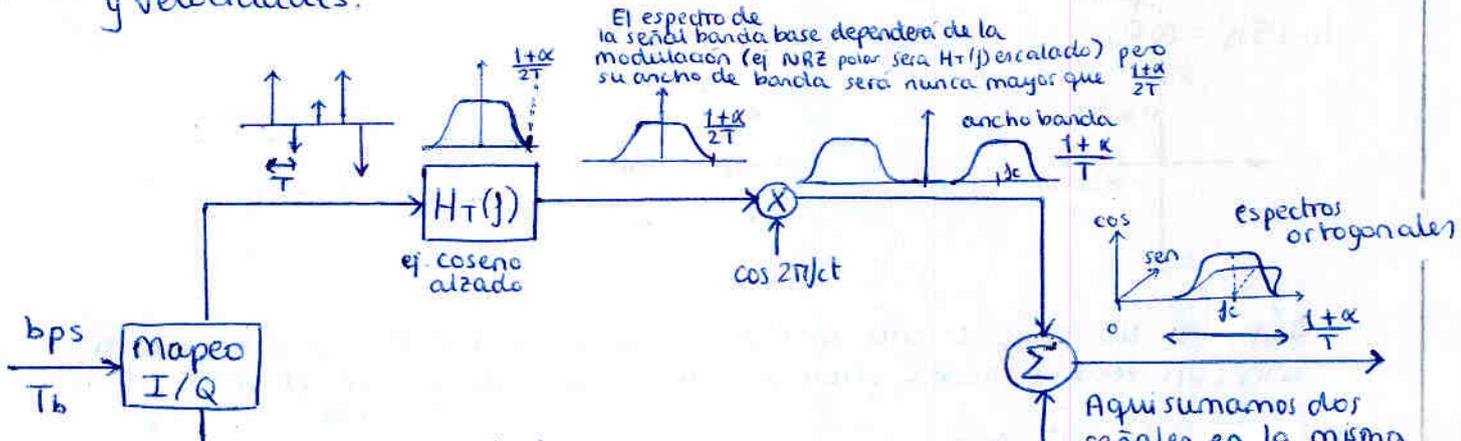
Es la suma de dos 6-ASK una usando $\cos(2\pi fct - \frac{\pi}{4})$ como portadora y la otra $\cos(2\pi fct + \frac{\pi}{4})$

Ademas hemos eliminado las esquinas. (Recuerda que la energía de una señal es su modulo al cuadrado) Queremos que sea lo mas circular posible.

Modulador Bidimensional



Algunas notas sobre periodos, anchos de banda y velocidades:



mapeador InPhase-Quadrature, Debe agrupar $\log_2 M$ bits para enviar la delta correspondiente a cada rama

(M: número total de símbolos a la salida del modulador)

por eso $T = \log_2 M \cdot T_b$

$$R_M = \frac{1}{T} = \frac{1}{\log_2 M} \cdot \frac{1}{T_b} = \frac{1}{\log_2 M} \cdot R_I$$

Por lo tanto, como cabía esperar: $R_I = \log_2 M \cdot R_M$

Aquí sumamos dos señales en la misma zona del espectro, pero no para nada, ya que son ortogonales.

El ancho de banda final será $\frac{1+\alpha}{T} = B$

$$\frac{1}{T} = R_M = \frac{B}{1+\alpha}$$

Nota: en el caso de $M = N^2$

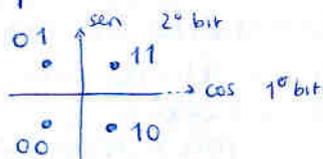
N será el número posible de niveles de deltas en cada una de las ramas i.e. el número de símbolos de las modulaciones unidimensionales de las cuales se obtiene la bidimensional.

Por tanto el Mapeo I/Q puede esperar $\log_2 N$ para enviar la delta por una rama y $\log_2 N$ para enviar la siguiente.

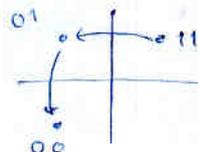
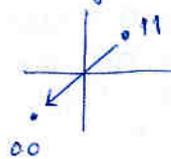
Aun así la separación entre deltas seguiría siendo $T = \log_2 M$

Sin embargo puede tener utilidad

ejemplo 4-PSK



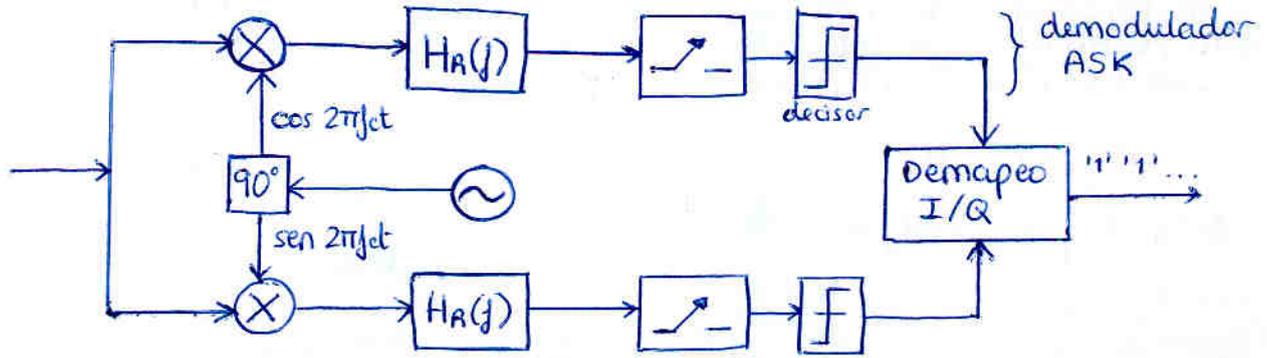
Si llega 1100 normalmente cambiaría directamente pero con la tecnica de arriba 1º cambiaría el 1º bit y luego el 2º



Luego veremos la utilidad de esto

Demodulador bidimensional

- Coherente

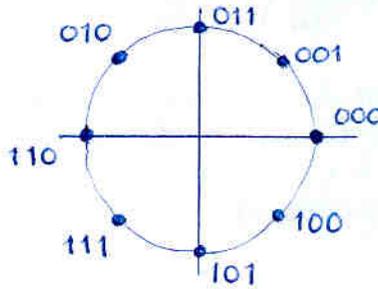


como ya sabemos, un desincronismo de fase "rota" la constelación

- No coherente: utiliza esquemas diferenciales

Codificación Gray

Para que un error con un simbolo continuo solo genere error de un bit.



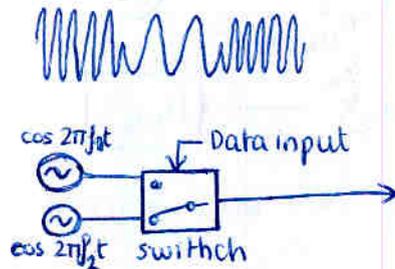
Modulaciones M-dimensionales

• Modulaciones FSK

Se transmite una frecuencia u otra (no vale combinarlas, eso recibe otro nombre) según el dato a transmitir.
Las frecuencias deben ser "ortogonales"

FSK no coherente

$$f_i - f_j = \frac{k}{T} \quad \text{Hay saltos de fase}$$



continuous phase FSK (CP-FSK)

no hay saltos de fase

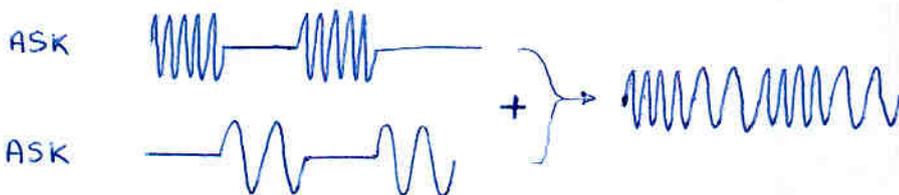


• FSK coherente

si las distintas frecuencias tienen mismo origen θ (FSK coherente)
(haciendo que el tiempo de los pulsos de data input sea un número entero de veces un periodo de los cosenos de distintas frecuencias)
Podemos pegar más las frecuencias

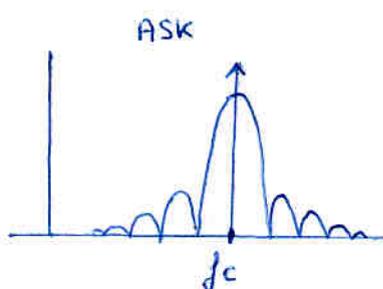
$$f_i - f_j = \frac{k}{2T}$$

Nota: Una FSK es una suma de M ASK unipolares (0,+V) de distintas frecuencias

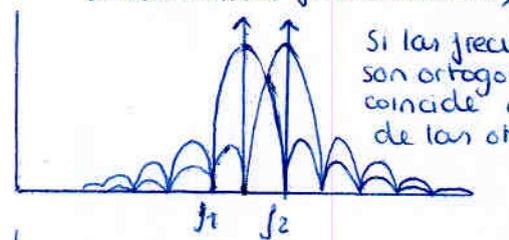


Esto es MUY util para reconocer el espectro

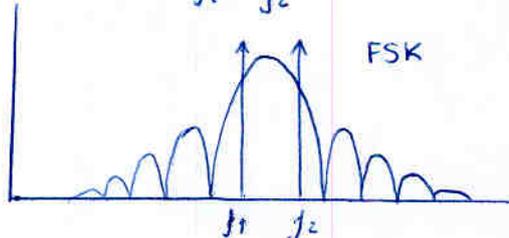
Espectro ASK (coseno alzado)



Espectro FSK (dos ASK sumandose a distintas frecuencias)

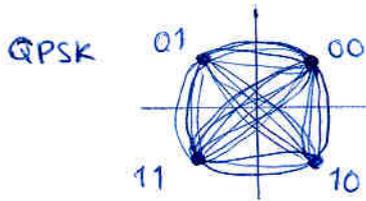


Si las frecuencias son ortogonales, cada una coincide con nulos de las otras



3.3 Otras modulaciones mas complejas

3.3.1. En la realidad, los cambios de un símbolo a otro no son instantáneos sino que llega de manera "continua"



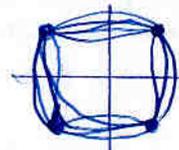
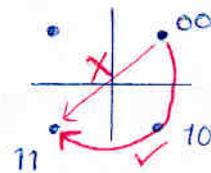
Esto implica que en ocasiones (ej: cambio de 180° en la fase) pasemos por cero (es decir, energía nula = envolvente nula)

Si esto ocurre, se agrava enormemente la aparición de armónicos donde no deberían estar, a causa de elementos no lineales. Esto podría afectar seriamente a modulaciones "contiguas" compartiendo el canal

Hay métodos ingeniosos para evitar esto:

• OQPSK (offset QPSK)

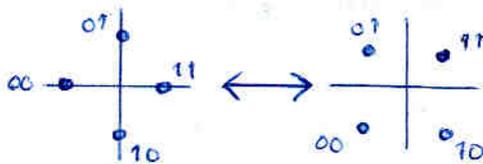
Cambiamos primero un bit y luego el otro. Ya vimos como se podía hacer esto en el mapeado IQ



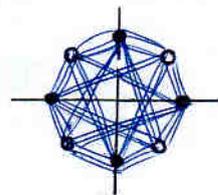
ya no se pasa por cero

• $\pi/4$ - DQPSK

Dos modulaciones Q-PSK que se van alternando en el tiempo



De esta forma el máximo cambio de fase es de 135°



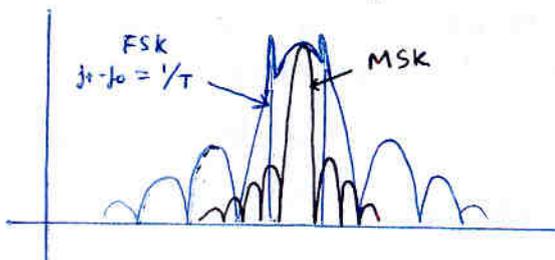
asi nunca pasamos por cero

3.3.3. MSK (Minimum Shift Keying) o FFSK (Fast FSK)

Es una CP-BFSK (continuous phase binary FSK) en la que exigiendo que ambos tonos empiecen con la misma fase hacemos

$$|f_1 - f_0| = \frac{1}{2T_b} \quad \text{Minima separación entre tonos distinguibles ortogonalmente}$$

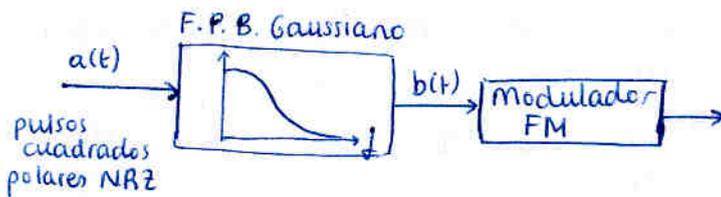
Reduciendo la separación reducimos el ancho de banda



Ademas una MSK es equivalente a una OQPSK (offset QPSK) pero simbolos en banda base sinusoidales

⇒ Sirven moduladores/demoduladores tanto de CP-BFSK como de O-QPSK

3.3.4. GMSK (Gaussian MSK)



Se logra un ancho de banda aún menor que con MSK (se usa en GSM)

Ancho de banda difícil de obtener analíticamente

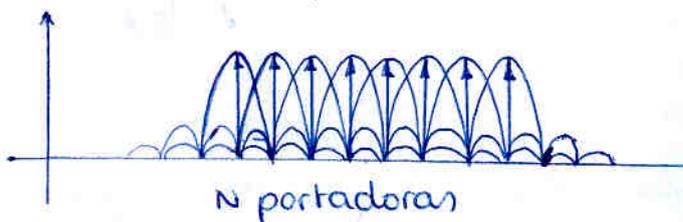
B · T _b	Ancho de banda (para cierto % de potencia) como fracción de R _T			
	90%	99%	99.9%	99.99%
0.2 GMSK	0.52	0.79	0.99	1.22
0.25 GMSK	0.57	0.86	1.09	1.37
0.5 GMSK	0.69	1.04	1.33	2.08
MSK	0.78	1.20	2.76	6.00

no entiendo la tabla

si $B \cdot T = \frac{B}{R_T} = 0.2 / 0.25 / 0.5$ porque luego dan $\frac{B}{R_T}$?????

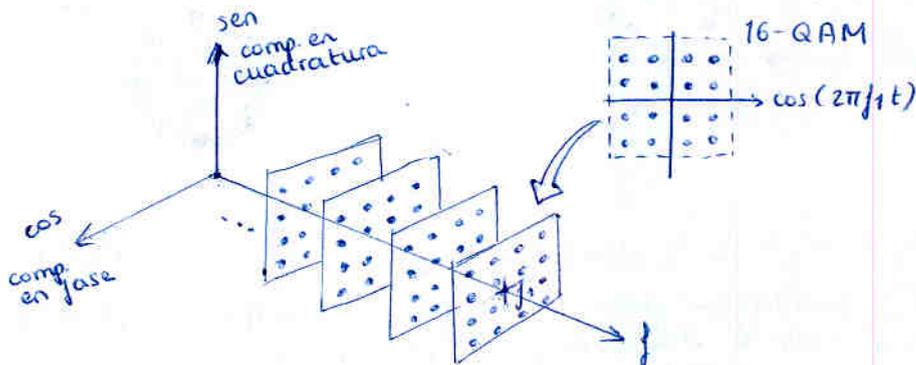
3.3.5 OFDM (Orthogonal Frequency Division Modulation)

Conjunto de portadoras separadas $\frac{1}{T}$ Hz (ortogonales) moduladas cada una de ellas (usualmente en M-QAM) a una velocidad de modulación $A_m = \frac{1}{T}$ Baud



se usará en TDT (TV digital terrestre)

otra forma de verlo es N M-QAM's a distintas frecuencias



Para cada simbolo, las N portadoras tendrán cada una cierta amplitud y cierta fase.

Problema: para modular/demodular necesitaremos N moduladores/demoduladores QAM sincronizados !!! → Impráctico

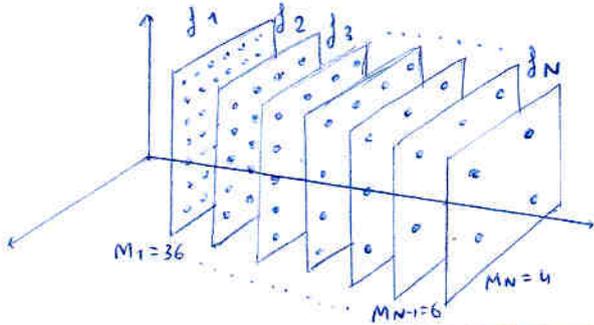
Solución: como cada portadora a frecuencia f_k es un tono con cierta amplitud y fase $s_k(t) = A_i (e^{j2\pi f_k t + \theta_i} + e^{-j2\pi f_k t + \theta_i})$

la señal total OFDM $S_{OFDM}(t) = \sum_{k=-N}^{+N} A_i e^{j2\pi f_k t + \theta_i}$

IFFT (Inverse Fast Fourier Transform) → con un SIMPLE DSP lo podemos demodular

Variación: DMT (Discrete MultiTone modulation)

Como las portadoras a mayor frecuencia se atenúan más (menor S/N) podemos "jugar" con el número M de símbolos que lleva cada una (i.e. número $\log_2 M$ de bits) para que las portadoras de altas frecuencias, que tienen más ruido, tengan menor número de niveles y por tanto mayor margen de ruido



Es el estándar de ADSL
(permite varios Mbit/s por par cables)

Será lo que use la TDT

$$V_T = \sum_{i=1}^N V_{T_i} = V_M \sum_{i=1}^N \log_2 M_i = \frac{1}{T} \sum_{i=1}^N \log_2 M_i$$

\uparrow velocidad transmisión [bits/s]
 \uparrow velocidad modulación [Bd]
 \uparrow periodo de símbolo

3.3.6 TCM (Trellis Coded Modulation)

Aumento estratégico del número de puntos en la constelación de señales añadiendo redundancia para corregir errores

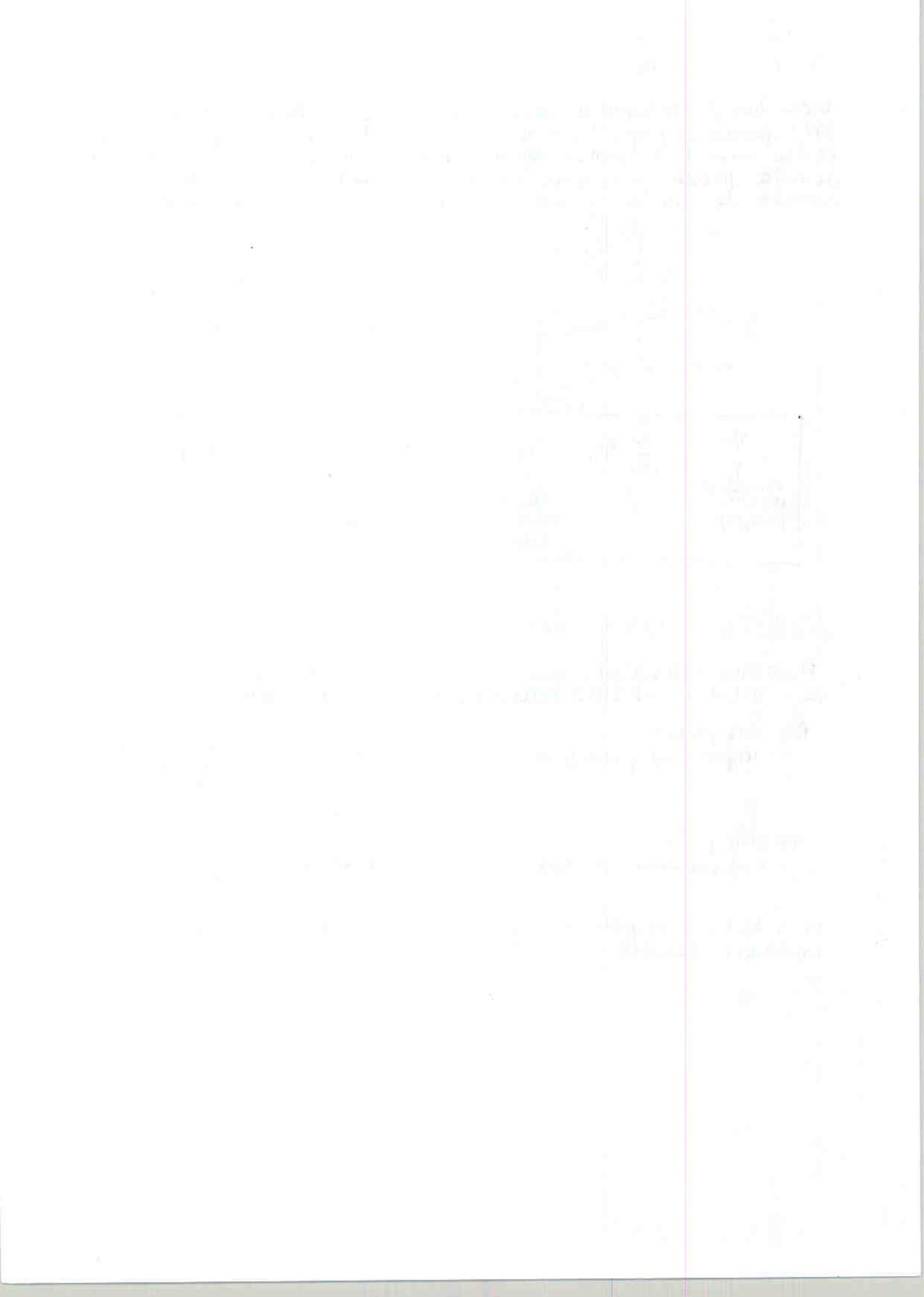
Por una parte:

- Mayor nº de puntos \rightarrow Incremento de P_B (si mantenemos la potencia, los puntos estarán más próximos)
 \uparrow probabilidad de error por bit

Por otra parte

- Redundancia \rightarrow corrección de errores \Rightarrow disminuye P_B

si el balance es positivo (garancia de codificación) nos interesará hacerlo.



Preguntas Test Tema 4

1. Filtro ideal paso bajo ancho de banda B
¿A que velocidad se puede transmitir sin ISI?

- a) sólo a $2B$ Baudios
- b) a submúltiplos de $2B$ Baudios ← ya que si se puede transmitir a $\frac{1}{T_b}$ se puede a $\frac{1}{2T}, \frac{1}{3T}, \frac{1}{4T}, \dots$
- c) cualquiera $\leq 2B$

2. Ancho de banda señal banda base con pulsos de tipo raíz cuadrada de Nyquist con $\alpha = 0.75$ y velocidad modulación $2k$ Baud

En un pulso de Nyquist $B = \frac{1+\alpha}{2T} = \frac{1+0.75}{2 \cdot \frac{1}{2k}}$

3. Ventajas que se obtienen con la precodificación en técnicas de señalización por respuesta parcial.

- a) d.e.p mas compacta
- b) Mayor (S/N) a entrada decodificador
- c) Menor probabilidad de error ← la precodificación impedia que hubiera arrastre de errores.
- d) Mayor velocidad de modulación

4 sea $p_1(t) \rightarrow P_1(f) = T_b \cdot \Pi\left(\frac{f}{f_b}\right) e^{-j\omega T_b}$
 $f_b = 1/T_b$

Si hacemos $p_2(t) = p_1(t) + p_1(t - T_b)$

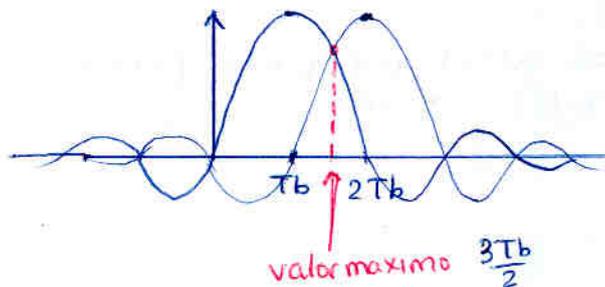
¿Cual es el valor máximo de $p_2(t)$?

$T_b \cdot \text{pulso cuadrado} \xrightarrow{TF^{-1}} \text{sinc}$
 mult por $e^{-j\omega T_b} \xrightarrow{TF^{-1}} \text{desplazamiento } +T_b$

por tanto:

$$p_1(t) = \text{sinc}\left(\frac{t - T_b}{T_b}\right)$$

$$p_2(t) = \text{sinc}\left(\frac{t - T_b}{T_b}\right) + \text{sinc}\left(\frac{t - 2T_b}{T_b}\right)$$



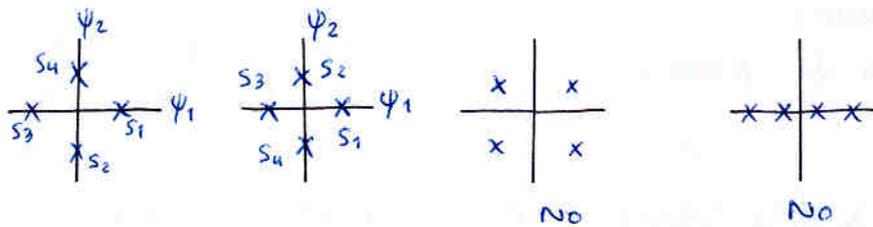
Valor máximo es $2 p_1\left(\frac{3T_b}{2}\right)$

recuerda $\text{sinc}(x) = \frac{\text{sen } \pi x}{\pi x}$

6. Constelación 4-PSK

$$S_i(t) = \cos(\omega_c t + \theta_i)$$

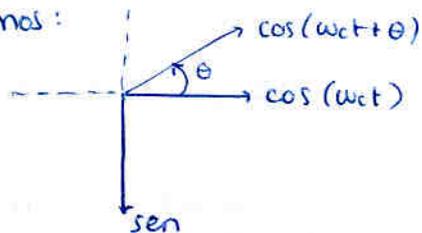
$$\theta_1 = 0, \quad \theta_2 = \pi/2, \quad \theta_3 = \pi, \quad \theta_4 = 3\pi/2$$



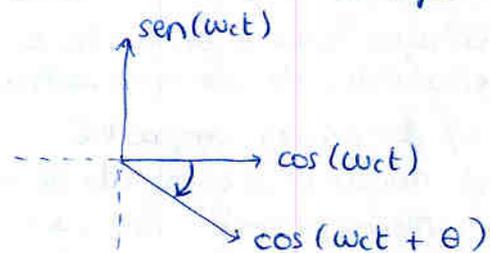
siendo $\psi_1 = \sqrt{\frac{2}{T}} \cos \omega_c t$

$$\psi_2 = \sqrt{\frac{2}{T}} \sin \omega_c t$$

Recordemos:



Pero como el sistema de referencia es



TRUCO recordar
 $\cos(\omega_c t - \frac{\pi}{2}) = \sin(\omega_c t)$
 $\cos(\omega_c t + \frac{\pi}{2}) = -\sin(\omega_c t)$

Por lo tanto el a) es la respuesta buena

7. Siendo canal ideal paso todo.

Para conseguir ISI nula es suficiente disponer

- Filtro de coseno alzado en emisor y receptor
- Filtro coseno alzado en receptor y raíz coseno alzado en receptor
- Ninguna de las anteriores

el filtro total $H_e(f) \cdot H_c(f) \cdot H_r(f)$
debe ser un coseno alzado

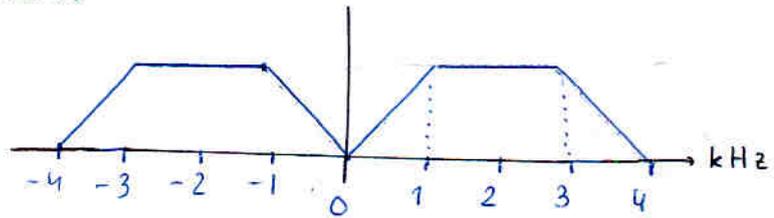
Forma habitual: utilizar raíz de coseno alzado en emisor y receptor

Y así encima tenemos un filtro adaptado para combatir el ruido.

8. En sistema de señalización por respuesta parcial

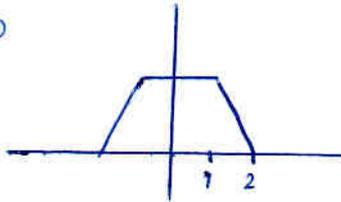
c) tenemos una IES conocida

6. Función de transferencia

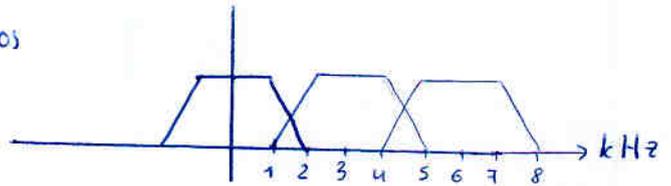


máxima velocidad de señalización sin IES

En el caso



Podemos



Si en este caso se puede 3 kHz también se puede en la de arriba (que es esta modulada con un coseno)

$$\frac{1}{T} = 3 \text{ kHz}$$

máxima velocidad de modulación

↳ máxima separación posible entre las replicas

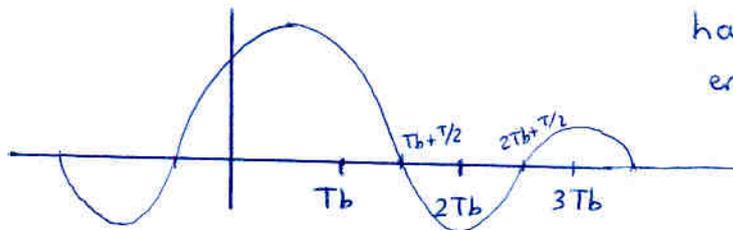
" máxima $\frac{1}{T} \Rightarrow$ mínimo T

7. sistema con respuesta frecuencial

$$P_1(f) = A \cdot \sqrt{L\left(\frac{1}{f_b}\right)} e^{-j\omega T/2} \quad f_b = \frac{1}{T_b}$$

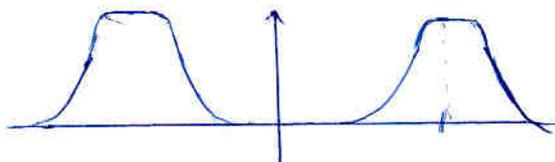
¿Cuál es el instante ideal de muestreo que no presente IES?

$p_1(t)$ es una sinc desplazada $T/2$



hay que muestrear en $t = \frac{T}{2} + kT_b$

8. ¿Cuántos bits por segundo acepta un modulator 16PSK con ancho de banda 690 Hz, usando pulsos de coseno atado con $\alpha = 0.15$?



Hemos utilizado que en PSK el espectro es igual que el de banda base (coseno atado) porque PSK es una suma de varias ramas de ASK (ortogonales) (una con seno y otra con coseno). Por eso el ancho de banda de la suma es el mismo.

$$= 2 \cdot \frac{1+\alpha}{2T} \Rightarrow \frac{1}{T} = \frac{690}{1.15} = R_m$$

$$= \frac{1+\alpha}{T} = 690$$

$$R_I = R_m \log_2 M$$

$$= \frac{690}{1.15} \log_2 4 = 2400 \text{ bit/s}$$

1. $y = x^2 + 2x + 1$



2. $y = x^2 - 4x + 4$

3. $y = x^2 - 6x + 9$



4. $y = x^2 - 8x + 16$



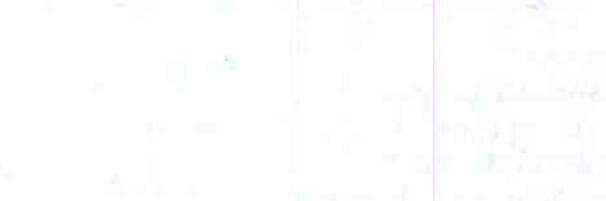
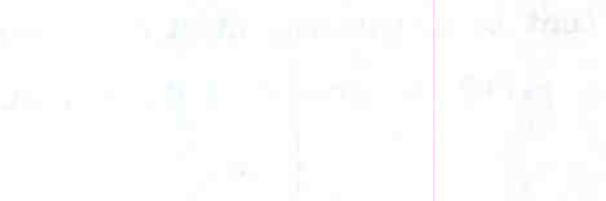
5. $y = x^2 - 10x + 25$



6. $y = x^2 - 12x + 36$



7. $y = x^2 - 14x + 49$



8. $y = x^2 - 2x + 1$

9. $y = x^2 - 4x + 4$

10. $y = x^2 - 6x + 9$

11. $y = x^2 - 8x + 16$

12. $y = x^2 - 10x + 25$

13. $y = x^2 - 12x + 36$

14. $y = x^2 - 14x + 49$

15. $y = x^2 - 16x + 64$

16. $y = x^2 - 18x + 81$

17. $y = x^2 - 20x + 100$

18. $y = x^2 - 22x + 121$

19. $y = x^2 - 24x + 144$

20. $y = x^2 - 26x + 169$

21. $y = x^2 - 28x + 196$

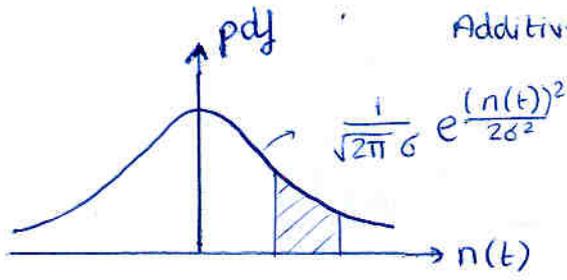
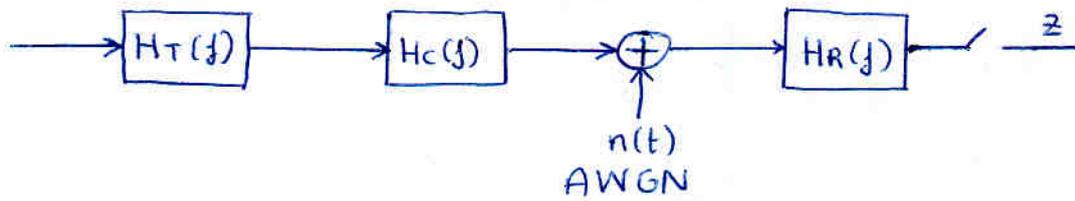
22. $y = x^2 - 30x + 225$

23. $y = x^2 - 32x + 256$

24. $y = x^2 - 34x + 289$

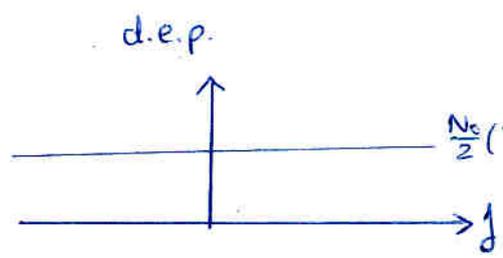
Tema 5. Canales

2. Canales AWGN



Additive white gaussian noise

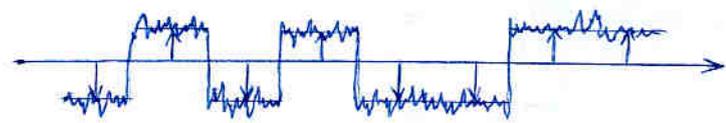
función densidad de probabilidad



densidad espectral de potencia

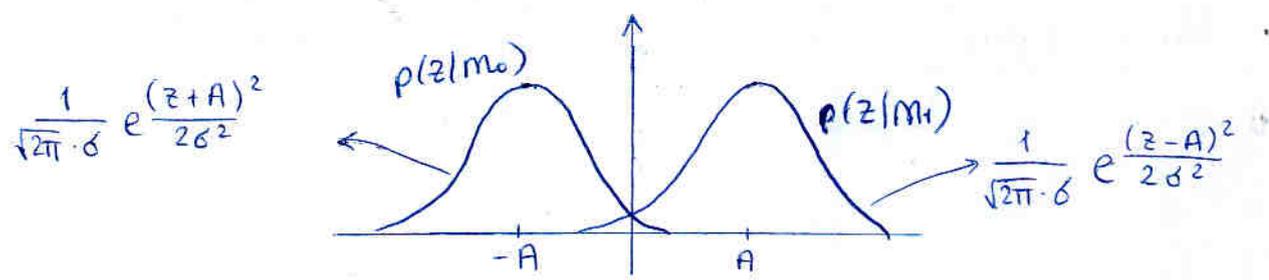
$$\int_{-\infty}^{\infty} G_n(f) df = \int_{-\infty}^{\infty} \frac{N_c}{2} df = \infty W \quad (\text{es un modelo; no tomarlo al pie de la letra})$$

Supongamos a la señal de $H_R(f)$ señal banda base siendo $p(t)$ pulsos cuadrados

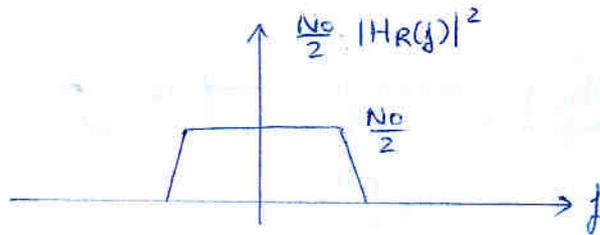


- El ruido a la salida de $H_R(f)$ ya no es blanco, es coloreado
- Se puede demostrar que una señal gaussiana sigue siendo gaussiana al atravesar un sistema lineal

Si dibujamos la densidad de probabilidad de los valores muestreados (z)



- Sabiendo que la potencia es la $\sigma^2 + \text{media}^2$
- la potencia del ruido en z será σ^2 (varianza)
- Si dibujamos la d.e.p. del ruido coloreado (tiene la forma de $|H_R(f)|^2$)



$$\text{Pot} = \sigma^2 = \int_{-\infty}^{\infty} \frac{N_0}{2} |H_R(f)|^2 df = \frac{N_0}{2} \int_{-\infty}^{\infty} |H_R(f)|^2 df = \frac{N_0}{2} \cdot E_{h_R(t)}$$

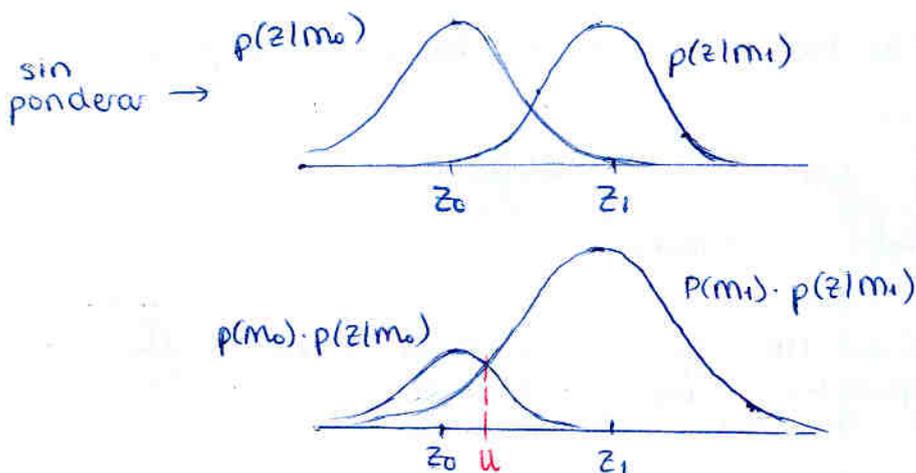
$$\Rightarrow \boxed{\text{Pot ruido} = \sigma^2 = \frac{N_0}{2} \cdot E_{h_R(t)}}$$

Veamos ahora el decisor:



Criterio MAP (máximo a posteriori)

consiste en ver dónde se cruzan las curvas de probabilidad ponderadas por la probabilidad del símbolo.



Criterio ML (maximum likelihood) (máxima verosimilitud)

El que seguiremos salvo que se indique lo contrario

$$\boxed{ML \Rightarrow P(m_0) = P(m_1) = 1/2}$$

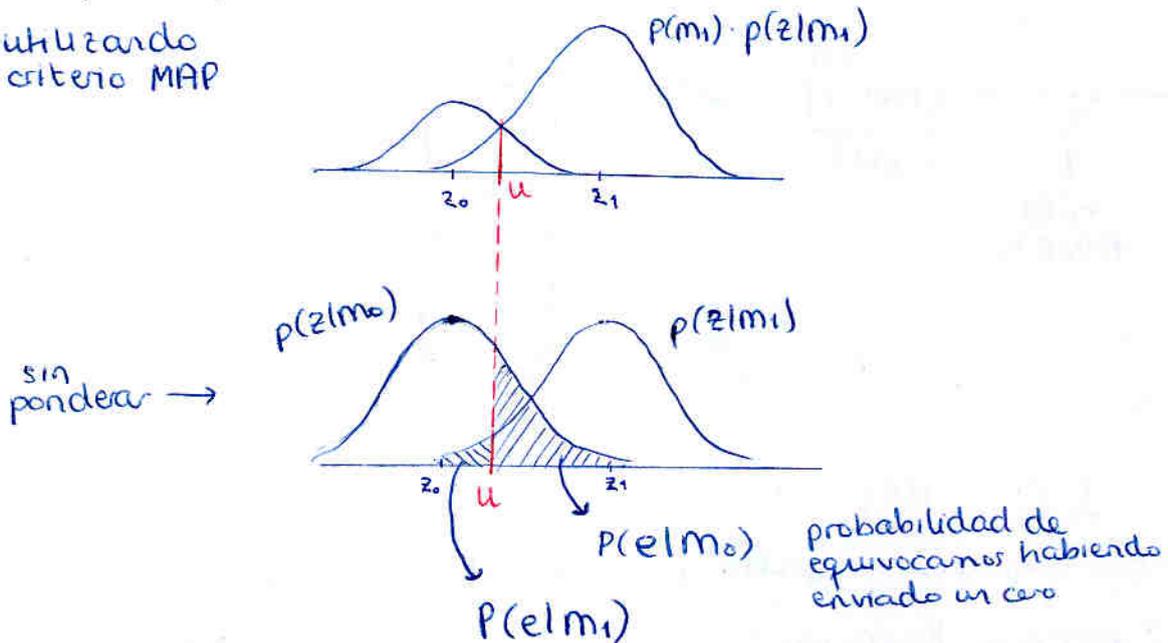
$$u = 0 \quad u = \frac{z_1 + z_0}{2}$$

cálculo de probabilidad de error

Tenemos 2 fuentes de error: enviar un cero o enviar un uno.
En ambos casos podemos equivocarnos.

Para mirar esa probabilidad, habría que mirar el área bajo la dep **sin ponderar**

ej: utilizando criterio MAP



$$P(e) = P(m_0) \cdot P(e|m_0) + P(m_1) \cdot P(e|m_1)$$

$$P(e|m_0) = \int_u^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-z_0)^2}{2\sigma^2}} dz = \int_{\frac{u-z_0}{\sigma}}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot \sigma \cdot dx$$

utilizamos

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$$

$$= Q\left(\frac{u-z_0}{\sigma}\right)$$

En el caso de ML $\Rightarrow P(m_0) = P(m_1)$

$$P(e) = \frac{1}{2} P(e|m_0) + \frac{1}{2} P(e|m_1)$$

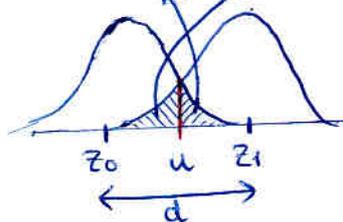
$$= \frac{1}{2} Q\left(\frac{u-z_0}{\sigma}\right) + \frac{1}{2} Q\left(\frac{z_1-u}{\sigma}\right)$$

y, como

$$Q\left(\frac{u-z_0}{\sigma}\right) = Q\left(\frac{z_1-u}{\sigma}\right) = Q\left(\frac{z_1-z_0}{2\sigma}\right) = Q\left(\frac{d}{2\sigma}\right)$$

$$= Q\left(\frac{d/2}{\sigma}\right)$$

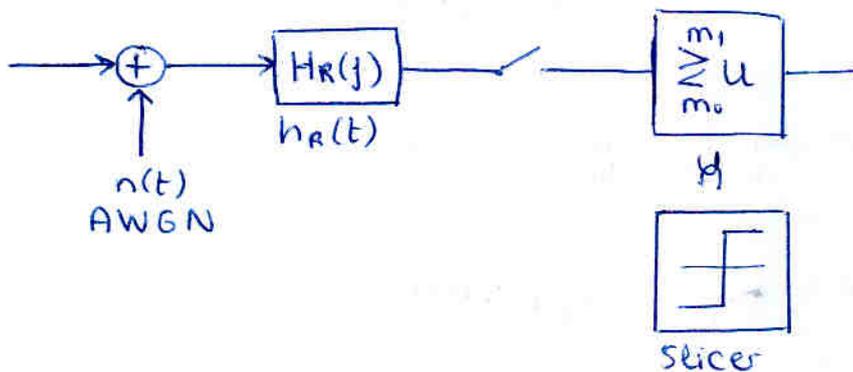
$u = \frac{z_1+z_0}{2}$



Obtenemos finalmente:

$$P(e) = Q(d/2\sigma)$$

- siendo $d = z_1 - z_0$
- estando Q tabulado
- siendo $p(m_0) = P(m_1)$
(a partir de ahora lo supondremos siempre) salvo que se indique lo contrario)



Filtro adaptado:

Filtro adaptado = filtro que maximiza la $(\frac{S}{N})$ a su salida

Se consigue haciendo:

$$h_R(t) = s_i^*(T-t)$$

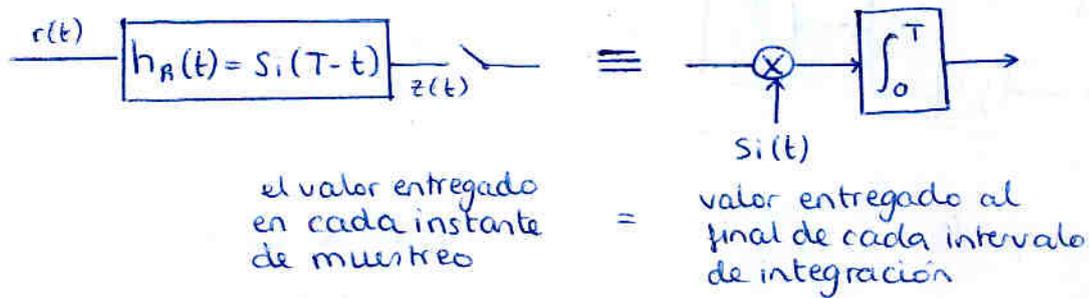
$s_i(t)$
señal del
símbolo
i-ésimo

- el símbolo invertido, desplazado (a efectos prácticos no importa) y conjugado (el símbolo es real, así que no afecta)

- Si $h_R(t)$ es raíz cuadrada de coseno alzado
Mato dos pájaros de un tiro

- ↳ Logro respuesta total de coseno alzado → NO ISI
(si transmito con $H_T(j)$ también de raíz)
(de coseno alzado)
- ↳ Logro filtro adaptado → minimizo $(\frac{S}{N})$

Filtro y correlador



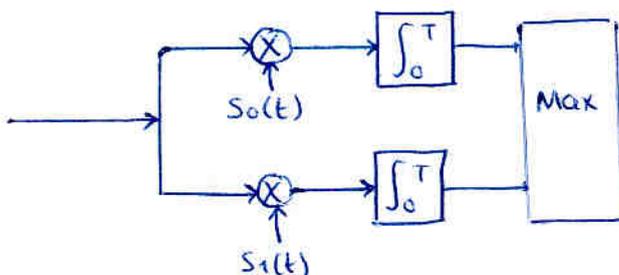
Dem:

$$\begin{aligned}
 z(t) &= r(t) * h_R(t) \\
 &= \int_{-\infty}^t r(\tau) \cdot h_R(t-\tau) d\tau \quad \left. \begin{array}{l} \\ \end{array} \right\} h_R(t) = S_i(T-t) \\
 &= \int_{-\infty}^t r(\tau) \cdot S_i(T-t+\tau) d\tau \\
 z(t) \Big|_{t=T} &= \int_{-\infty}^T r(\tau) \cdot S_i(T-\tau+\tau) d\tau \\
 &= \int_{-\infty}^T r(\tau) \cdot S_i(\tau) d\tau \quad \left. \begin{array}{l} \\ \end{array} \right\} S_i(\tau) \in [0, T] \\
 &= \int_0^T r(\tau) \cdot S_i(\tau) d\tau
 \end{aligned}$$

Además recordemos que el correlador era un extractor de coordenadas (proyectaba la señal en el eje $S_i(t)$)

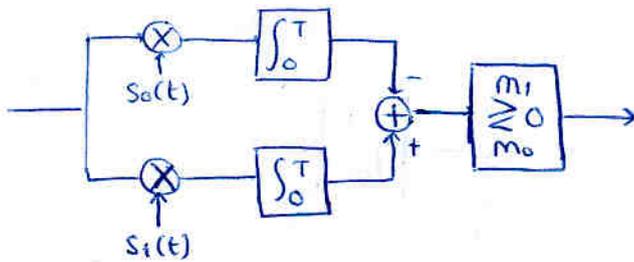
Por tanto podremos representar el demodulador como un filtro lineal $h_R(t)$, ya que es equivalente a un extractor de coordenadas.

$$S_i(t) = \begin{cases} S_0(t) = a_0 \cdot p(t) \\ S_1(t) = a_1 \cdot p(t) \end{cases}$$



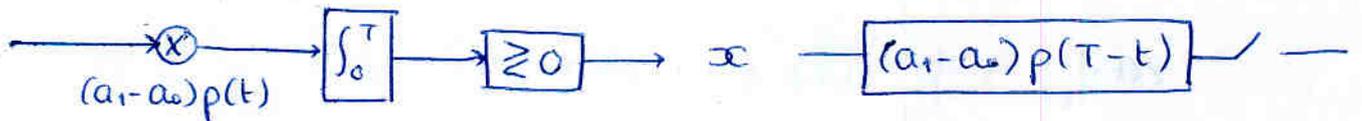
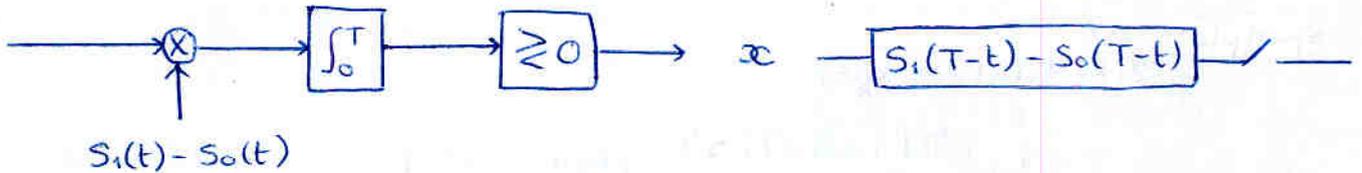
La señal que sea mayor será la que gane.

una forma mejor sería

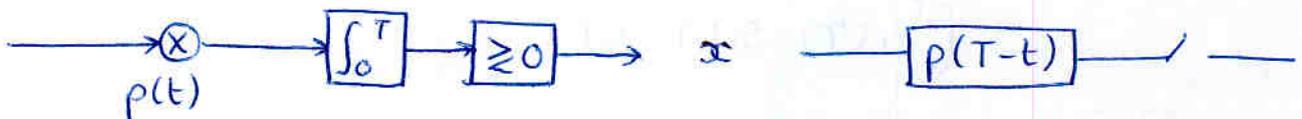


o, aún mejor !!

es equivalente a:

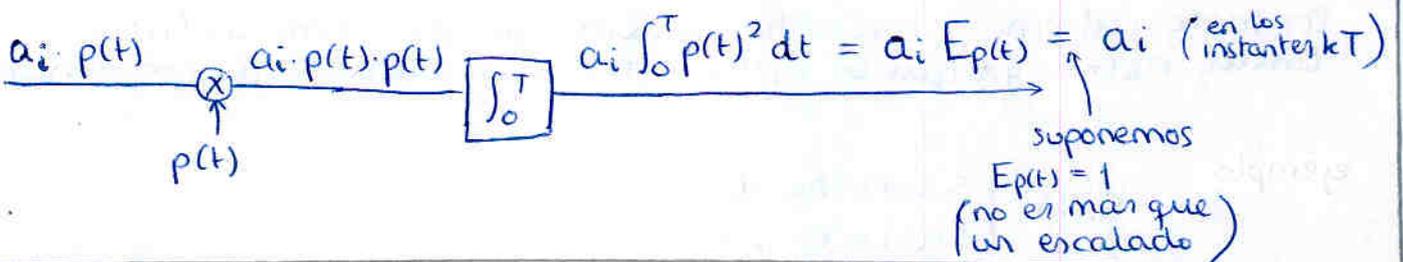


y, ya que $(a_1 - a_0)$ no es más que una constante



comprobación:

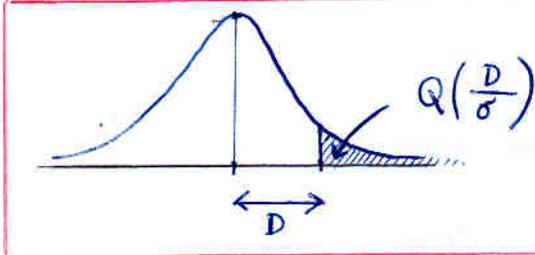
si enviamos $S_o(t) = a_i \cdot p(t)$



Probabilidad de error de símbolo P_e

También llamada Probabilidad de Error a secas

Distribución Gaussiana



En el caso de ruido gaussiano

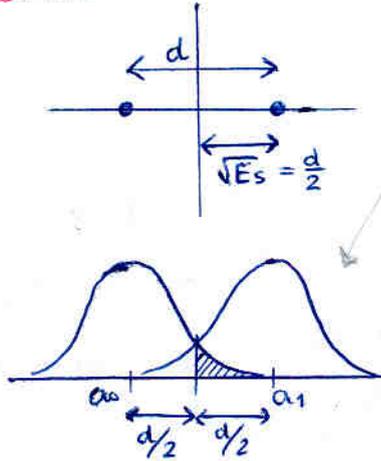
$$\sigma^2 = \frac{N_0}{2}$$

$$\sigma = \sqrt{N_0/2}$$

$$\frac{N_0}{2} \left(\frac{W}{Hz} \right) \rightarrow \boxed{HR(f)} \rightarrow \frac{N_0}{2} \int_{-\infty}^{\infty} |HR(f)|^2 df$$

para espacio ortonormal

BPSK



$$P_e = \frac{1}{2} \cdot P(e|s_0) + \frac{1}{2} \cdot P(e|s_1)$$

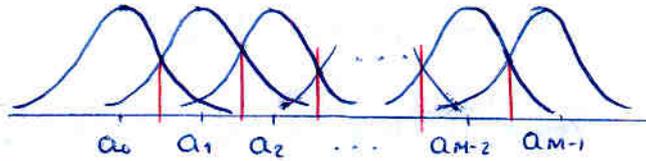
$$= \frac{1}{2} \cdot Q\left(\frac{d/2}{\sigma}\right) + \frac{1}{2} \cdot Q\left(\frac{d/2}{\sigma}\right)$$

$$= Q\left(\frac{d}{2\sigma}\right) //$$

y como $d = 2\sqrt{E_s}$ y $\sigma = \sqrt{N_0/2}$

$$= Q\left(\frac{2\sqrt{E_s}}{2\sqrt{N_0/2}}\right) = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) // = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Unidimensional



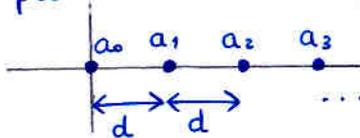
símbolos externos $P(e|w_0) = P(e|w_{M-1}) = Q\left(\frac{d}{2\sigma}\right)$

símbolos internos $P(e|w_i)_{i \neq 0, M-1} = 2 \cdot Q\left(\frac{d}{2\sigma}\right)$

$$P_e = \frac{2}{M} Q\left(\frac{d}{2\sigma}\right) + \frac{M-2}{M} \cdot 2Q\left(\frac{d}{2\sigma}\right)$$

Ahora solo falta escribir d en función de E_s

ejemplo



$$E_s = \frac{1}{M} \sum_{i=0}^{M-1} E_i = \frac{1}{M} \sum_{i=0}^{M-1} (i \cdot d)^2 = \frac{d^2}{M} \sum_{i=0}^{M-1} i^2$$

despejando $d = \sqrt{\frac{M \cdot E_s}{\sum_{i=0}^{M-1} i^2}}$

NOTA:

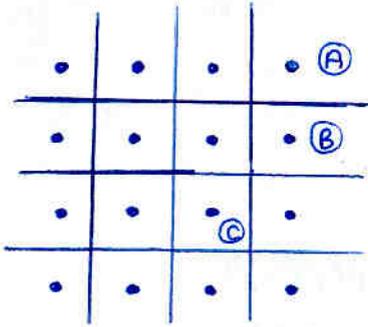
$$\sum_{i=0}^{M-1} i^2 = \frac{M \cdot (M-1) \cdot (2M-1)}{6}$$

La expresión será válida ^{también} en el caso de base ortogonal (no ortonormal) donde, aunque $E_i \neq \|a_i\|^2$ sino que $E_i = E_{\psi_i} \cdot \|a_i\|^2$, el escalado afecta también a σ .

Bidimensional

- Regiones de decisión rectangulares

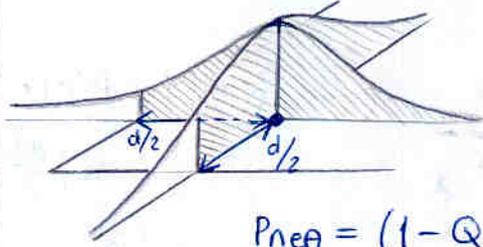
ej 16 QAM



Hay 3 tipos de regiones

$$\textcircled{A} \quad P_{eA} = 1 - P_{\text{no errores en A}}$$

Para calcular $P_{\text{no errores en A}}$ tomamos como origen el punto



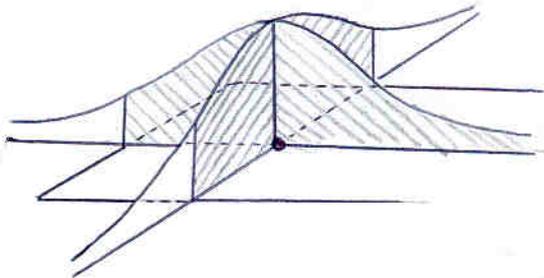
$$P_{\text{no errores en A}} = (1 - Q(\frac{d}{2\sigma})) \cdot (1 - Q(\frac{d}{2\sigma}))$$

$$\begin{aligned} P_{eA} &= 1 - P_{\text{no errores en A}} \\ &= 1 - (1 - Q)(1 - Q) \\ &= 2Q - Q^2 \end{aligned}$$

denotamos $Q := Q(\frac{d}{2\sigma})$ por simplicidad

es lógico es hacer $x < \frac{d}{2}$ OR $y < \frac{d}{2}$ MENOS la intersección ($x < \frac{d}{2}$ AND $y < \frac{d}{2}$) que habíamos contado doble

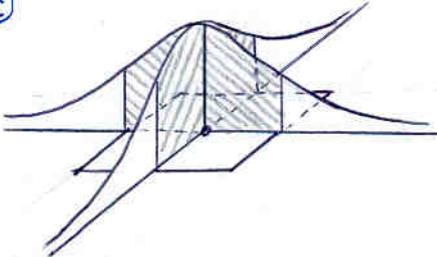
$$\textcircled{B} \quad P_{eB} = 1 - P_{\text{no errores en B}}$$



$$P_{\text{no errores en B}} = (1 - Q) \cdot (1 - 2Q)$$

$$P_{eB} = 1 - (1 - Q) \cdot (1 - 2Q) = 3Q - 2Q^2$$

\textcircled{C}

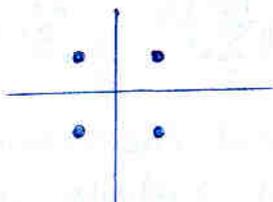


$$\begin{aligned} P_{eC} &= 1 - P_{\text{no errores en C}} \\ &= 1 - (1 - 2Q)^2 = 4Q - 4Q^2 \end{aligned}$$

La P_e total será

$$\begin{aligned} P_e &= \frac{4}{16} P_{eA} + \frac{8}{16} P_{eB} + \frac{4}{16} P_{eC} \\ &= \dots = 3Q - \frac{9}{4} Q^2 \end{aligned}$$

ej QPSK

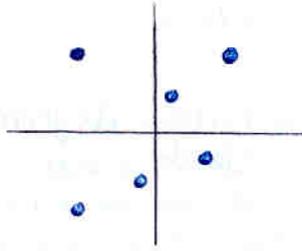


$$\begin{aligned} P_{eA} &= 1 - P_{\text{no errores en A}} \\ &= 1 - (1 - Q)^2 \\ &= 2Q - Q^2 \end{aligned}$$

$$P_e = \frac{4}{4} P_{eA}$$

- Regiones de decisión cualquiera

Teorema de Arthur y Dym



d_i : distancia del símbolo i -ésimo a su vecino más próximo.

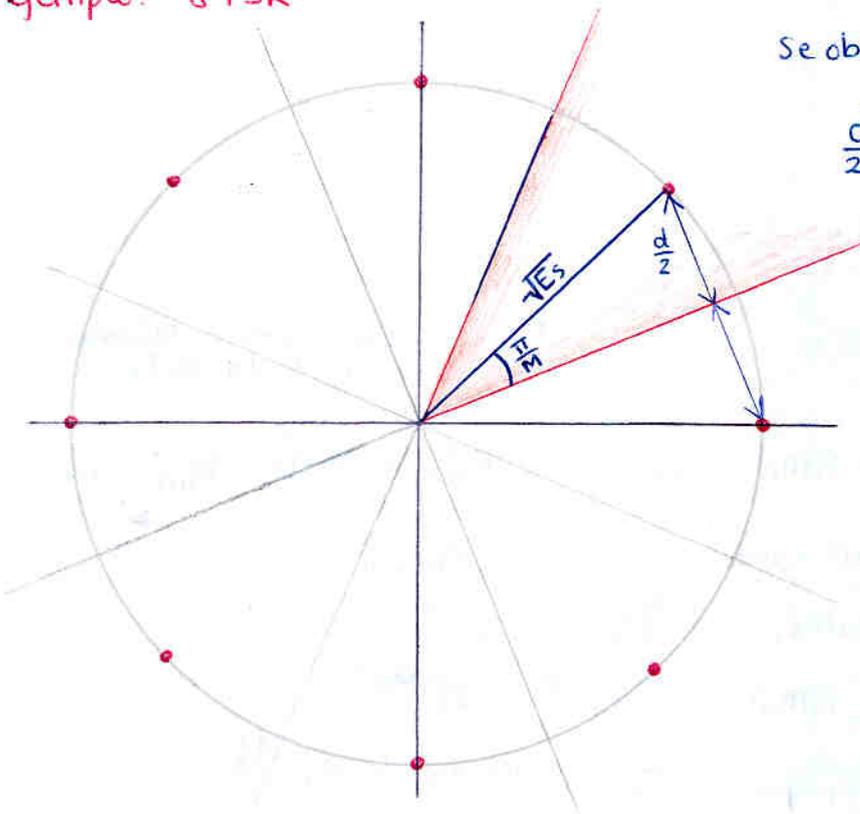
\bar{d} : media de d_i ($\frac{1}{M} \sum d_i$)

d^* : mínimo de los d_i

se cumple:

$$Q\left(\frac{\bar{d}}{2\sigma}\right) \leq P_e \leq (M-1) Q\left(\frac{d^*}{2\sigma}\right) \quad \text{NOTA } \sigma = \sqrt{2N_0}$$

ejemplo: 8 PSK



Se obtiene $\frac{d}{2}$ geoméricamente

$$\frac{d}{2} = \sqrt{E_s} \cdot \sin\left(\frac{\pi}{M}\right)$$

por tanto

$$d_i = 2\sqrt{E_s} \cdot \sin\left(\frac{\pi}{M}\right) \quad \forall i$$

$$\bar{d} = d_i$$

$$d^* = d_i$$

sustituyendo en el teorema de Arthur y Dym

$$Q\left(\frac{\bar{d}}{2\sigma}\right) \leq P_e \leq (M-1) Q\left(\frac{d^*}{2\sigma}\right)$$

$$Q\left(\frac{d}{\sqrt{2N_0}}\right) \leq P_e \leq (M-1) Q\left(\frac{d}{\sqrt{2N_0}}\right)$$

$$Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right) \leq P_e \leq (M-1) Q\left(\sqrt{\frac{2E_s}{N_0}} \sin \frac{\pi}{M}\right)$$

• Modulaciones tridimensionales

Hay volúmenes de decisión

Probabilidad de error de bit P_b

La P_e ; prob error de símbolo, implicaba equivocarse de símbolo: pero ya que un símbolo lleva $\log_2 M$ bits ¿En cuantos bits nos equivocamos cada vez que nos equivocamos de símbolo? i depende de con que símbolo nos equivocamos!

Se utiliza Codificación Gray al asignar bits a los símbolos, de forma que los símbolos más cercanos (a los que es más probable equivocarse) sólo se diferencian en un bit y por tanto la mayoría de veces nos equivocaremos únicamente en 1 bit de los $\log_2 M$ que tiene el símbolo.

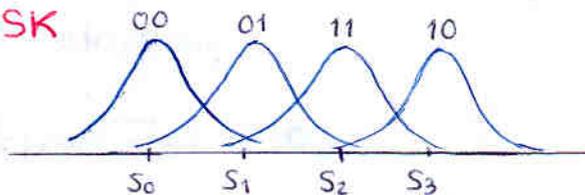
cuando no es abordable calcular P_b de forma exacta se usa:

- Código Gray
 - suponiendo que al equivocarnos de símbolo nos equivocamos siempre con el adyacente
- $$P_b \approx \frac{P_e}{\log_2 M}$$

Cálculo de P_b de forma exacta

- Unidimensional:

ej: 4-ASK

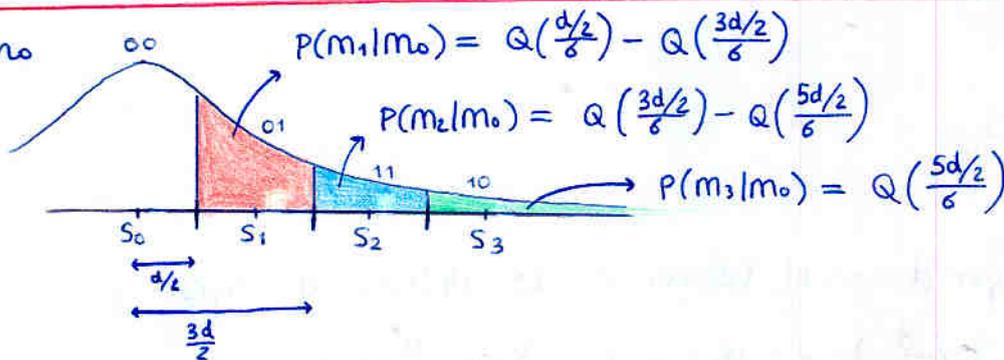


prácticamente idéntico al cálculo de P_e

$$P_b = P(m_0) \cdot P_b|_{m_0} + P(m_1) \cdot P_b|_{m_1} + P(m_2) \cdot P_b|_{m_2} + P(m_3) \cdot P_b|_{m_3} \quad (1)$$

El gran cambio y la dificultad está en calcular $P_b|_{m_i}$

ej: $P_b|_{m_0}$



se pondera equivocarse a cada símbolo por el n° de bits distintos partido el n° total de bits.

$$P_b|_{m_0} = \frac{1}{2} \cdot P(m_1|m_0) + \frac{2}{2} P(m_2|m_0) + \frac{1}{2} P(m_3|m_0)$$

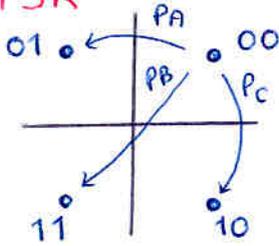
↑
nos equivocamos en 1 de los 2 bits que tiene el símbolo

Y ahora ya (haciendo lo de arriba para los 4) aplicamos la fórmula (1)
 $(P_b|_{m_0}, P_b|_{m_1}, P_b|_{m_2}, P_b|_{m_3})$

Bidimensional

- con regiones rectangulares

ej QPSK

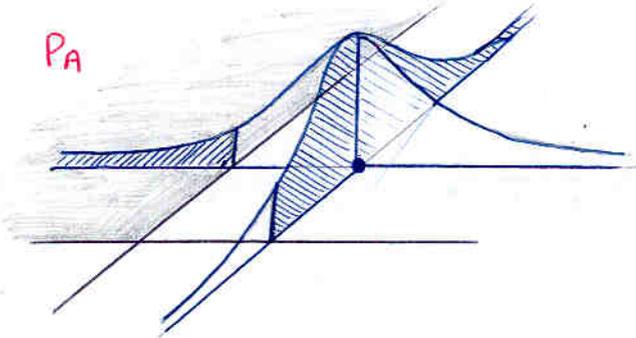


Como los 4 son simétricas;

$$P_b = P_{b|mo}$$

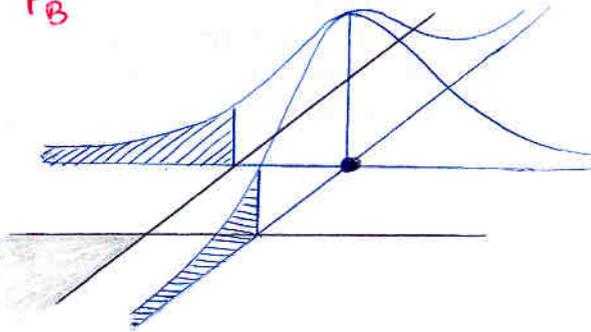
$$P_{b|mo} = \frac{1}{2} P_A + \frac{2}{2} P_B + \frac{1}{2} P_C$$

P_A



$$P_A = Q \cdot (1 - Q)$$

P_B

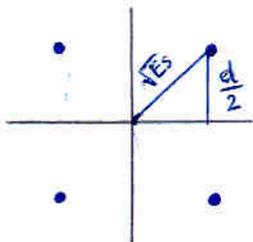


$$P_B = Q \cdot Q$$

$$P_C = P_A$$

Por lo tanto $P_b = P_{b|mo} = \frac{1}{2} P_A + \frac{2}{2} P_B + \frac{1}{2} P_C = \dots = Q \left(\frac{d}{2\sigma} \right)$

Ahora queremos d en función de E_s



$$E_s = \left(\frac{d}{2} \right)^2 + \left(\frac{d}{2} \right)^2 = \frac{d^2}{2}$$

$$d = \sqrt{2E_s}$$

y como $\sigma = \sqrt{\frac{N_0}{2}}$
sustituyendo:

$$P_b = Q \left(\frac{\sqrt{2E_s}}{2\sqrt{\frac{N_0}{2}}} \right) = Q \left(\sqrt{\frac{E_s}{N_0}} \right) = Q \left(\sqrt{\frac{2E_b}{N_0}} \right)$$

Para comparar con justicia
usar E_b

$$E_s = \log_2 M \cdot E_b$$

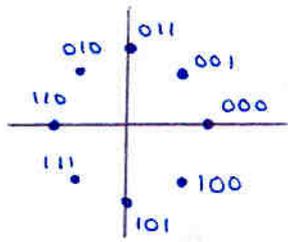
Nota:

- QPSK y BPSK tienen misma P_b .
- QPSK tiene más eficiencia espectral.
- Única desventaja de QPSK frente a BPSK \Rightarrow Requiere 2 ramas en el demodulador.

Nota: BPSK

$$M=1 \Rightarrow P_b = P_e = Q \left(\sqrt{\frac{2E_b}{N_0}} \right)$$

• con regiones por sectores



Es inabordable de calcular
se utiliza aproximación:

$$P_b \approx \frac{P_e}{\log_2 M}$$

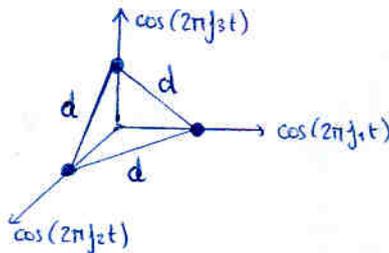
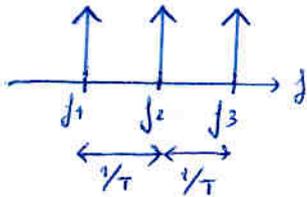
• M-dimensionales (M > 2)

$$k = \text{n}^\circ \text{ de bits} = \log_2 M$$

• simbolos equidistantes
(y equiprobables)

$$\Rightarrow P_b = \frac{M/2}{M-1} P_e = \frac{2^{k-1}}{2^k - 1} P_e$$

Dem:



Aunque en el espectro
no parecen equidistantes,
en la representación
M-dimensional se ve que
sí lo son.

Formas de equivocarnos en 1 bit: $\binom{3}{1} = 3$

Formas de equivocarnos en 2 bits: $\binom{3}{2} = 3$

Formas de equivocarnos en 3 bits: $\binom{3}{3} = 1$

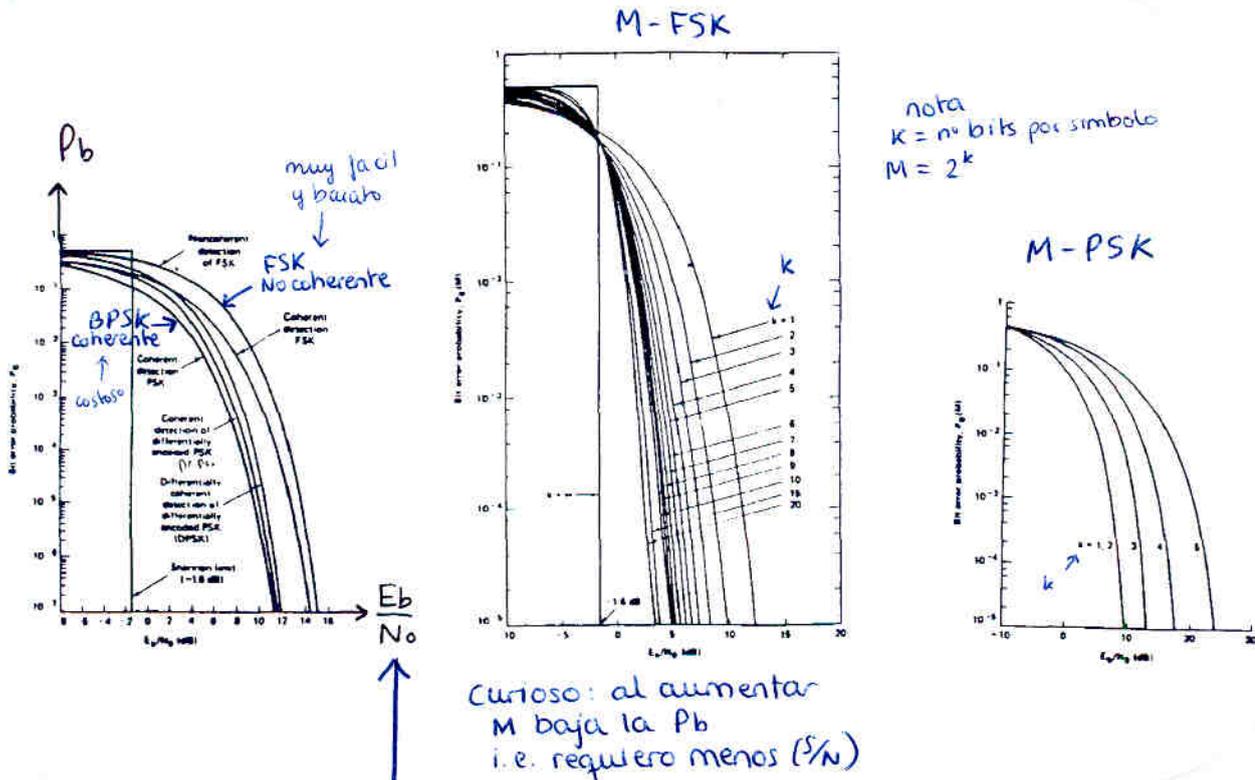
Ponderamos estas posibilidades por el n° de bits erróneos

$$P_b = \frac{1}{3} \cdot \binom{3}{1} \cdot P_e + \frac{2}{3} \cdot \binom{3}{2} \cdot P_e + \frac{3}{3} \cdot \binom{3}{3} \cdot P_e$$

Nota $\sum_{k=0}^n \binom{n}{k} = 2^n$

créate el resto

2.1.4. Gráficas de Pb: probabilidad de error de bit



Relación señal a ruido normalizada respecto a bit

Para comparar equitativamente

$$\frac{E_s}{N_0} = \log_2 M \frac{E_b}{N_0}$$

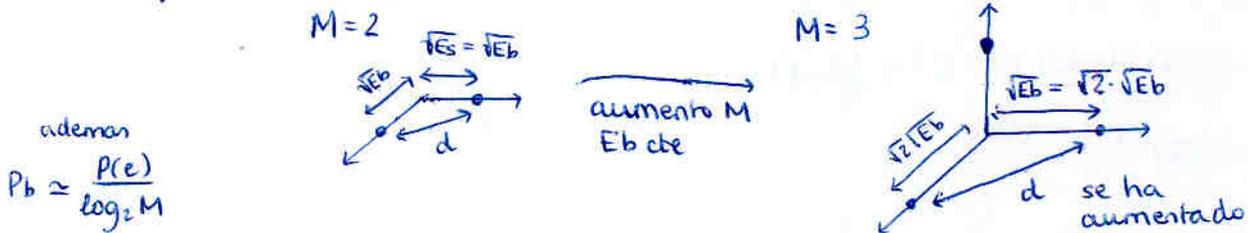
usamos esto

M-FSK: Al aumentar M, si E_b es cte \rightarrow P_b disminuye

Al aumentar M, aumentamos el número de bits por simbolo; si la energía por bit permanece igual implica que los símbolos, al llevar más bits, tendrán más energía

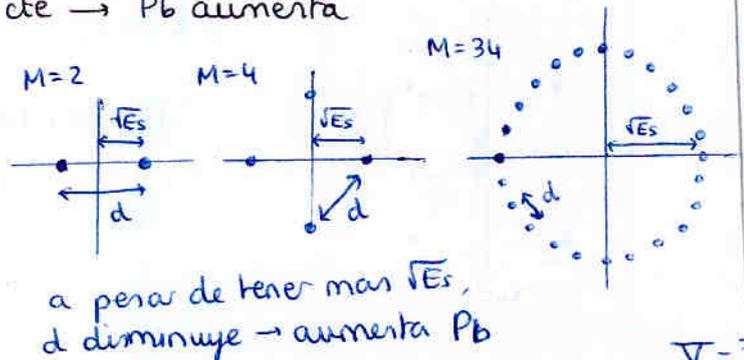
de hecho $E_s = \log_2 M \cdot E_b$

Aumentar E_s implica aumentar el módulo de las coordenadas, y como los nuevos símbolos aparecen en nuevas dimensiones, no se apelotonan; por tanto aumenta la distancia entre símbolos



M-PSK: Al aumentar M, si E_b es cte \rightarrow P_b aumenta

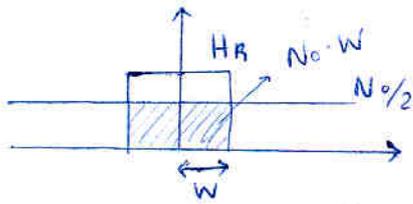
Aunque aquí también ocurre que la E_s aumenta con M, esto no compensa con el hecho de que añadir más símbolos los aglutina en sus 2 dimensiones



Relación entre $(\frac{S}{N})$ y $\frac{E_b}{N_0}$:

$$SNR = (\frac{S}{N}) = \frac{E_s/T_s}{N_0 W} = \frac{E_b \log_2 M}{N_0 W T_s} = \frac{E_b \log_2 M}{N_0 \cdot W \cdot T_s} = \frac{E_b \cdot \log_2 M}{N_0 \cdot W} R_M$$

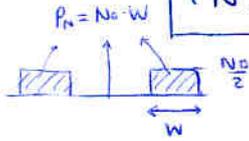
velocidad modulación



$$P_s = \frac{E_b \cdot R_I}{(P_b) \cdot (P_s)}$$

$$\left(\frac{S}{N}\right) = \frac{E_b}{N_0} \cdot \frac{R_I}{W}$$

$R_I \rightarrow$ tasa binaria
 $W \rightarrow$ ancho banda



para coseno alzado: $W = \frac{1+\alpha}{T}$
 $R_I = \frac{1}{T} \log_2 M$

$$\left(\frac{S}{N}\right) = \frac{E_b}{N_0} \cdot \frac{\log_2 M}{1+\alpha} \xrightarrow{\alpha=0 \text{ ideal}} \frac{E_b}{N_0} = \frac{(S/N)}{\log_2 M}$$

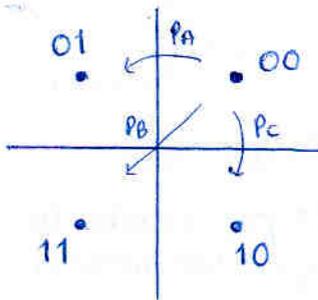
suponiendo H_R cuadrado
ejercicio

Cálculo de P_b exacta modulación Bidimensional QPSK

La fórmula aproximada, en la cual suponemos que nos equivocamos siempre con el símbolo contiguo

$$P_b = \frac{P_e}{\log_2 M}$$

Hagámoslo ahora de manera exacta para QPSK

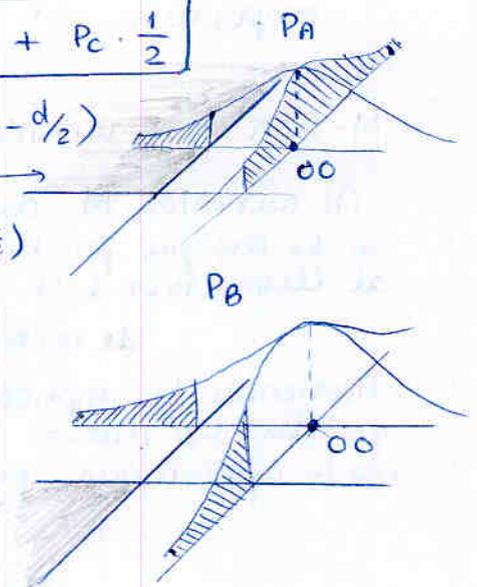


$$P_b = P_A \cdot \frac{1}{2} + P_B \cdot \frac{2}{2} + P_C \cdot \frac{1}{2}$$

$$P_A = P(n_x < -d/2, n_y > -d/2) = Q \cdot (1-Q)$$

$$P_B = P(n_x < -d/2, n_y < -d/2) = Q \cdot Q$$

$$P_C = P_A$$



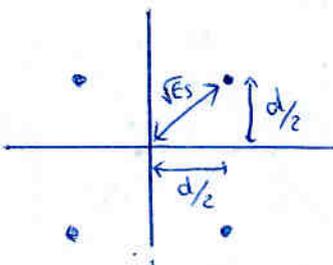
Por lo tanto

$$P_B = \frac{1}{2} Q \cdot (1-Q) + \frac{2}{2} \cdot Q^2 + \frac{1}{2} Q \cdot (1-Q)$$

$$P_B = Q \left(\frac{d}{2\sigma} \right)$$

Calculando E_s para despejar d :

$$E_s = \left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\right)^2 = \frac{d^2}{2} \rightarrow d = \sqrt{2E_s}$$



Por tanto

$$P_b = Q \left(\frac{\sqrt{2E_s}}{2\sigma} \right)$$

$$P_b = Q\left(\frac{\sqrt{2E_s}}{2\sigma}\right) \underset{\sigma^2 = \frac{N_0}{2}}{=} Q\left(\frac{\sqrt{2E_s}}{2\sqrt{\frac{N_0}{2}}}\right) = Q\left(\sqrt{\frac{E_s}{N_0}}\right)$$

y como $E_s = \log_2 M \cdot E_b = 2 E_b$

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad \text{QPSK}$$

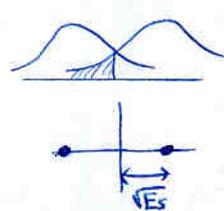
Recuerda:

AWGN \rightarrow $H_R(f)$ \rightarrow

$$G_{H(f)} = \frac{N_0}{2} \text{ (W/Hz)} \quad \sigma^2 = \frac{N_0}{2} \cdot \int_{-\infty}^{\infty} |H_R(f)|^2 df$$

$= \frac{N_0}{2} \cdot E_{HR} = 1$
para espacio ortonormal

Recuerda: BPSK



$$P_b = Q\left(\frac{d/2}{\sigma}\right) = Q\left(\frac{\sqrt{2E_b}}{2\sigma}\right)$$

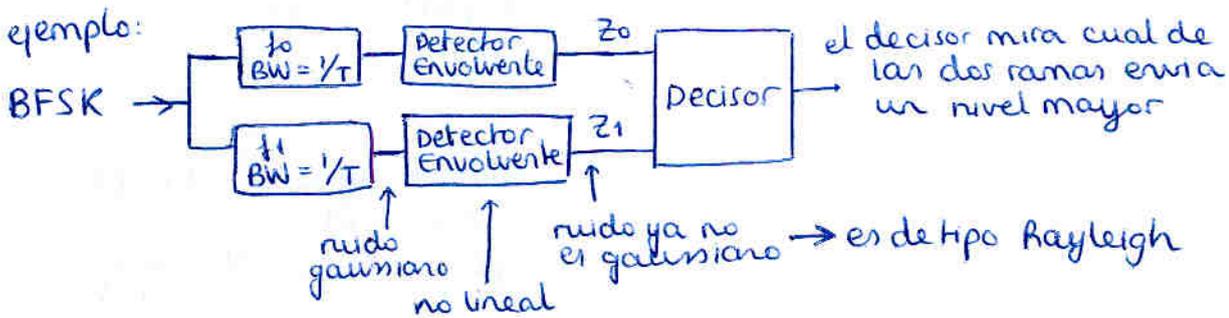
$$= Q\left(\frac{\sqrt{E_b}}{\sqrt{N_0/2}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

\Rightarrow QPSK tiene la misma P_b que BPSK a pesar de que QPSK tiene mas eficiencia espectral

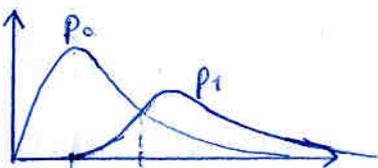
La UNICA desventaja de QPSK frente a BPSK es que requiere dos ramas en el receptor

Probabilidad de error en receptores no coherentes

No sirven las fórmulas anteriores. En estos receptores no necesitamos la fase y ademas hay elementos no lineales, por lo tanto el ruido deja de ser blanco gaussiano \rightarrow no sirven las fórmulas!!



Distribución Rayleigh



$$P(e|m_1) = P(z_0 > z_1 | m_1)$$

se comprueba que P_e es mayor que en receptor coherente

2.3. Compromisos de diseño en los esquemas de modulación

2.3.1 Limite de Shannon

Para un canal continuo AWGN, la capacidad C del canal determina el regimen binario libre de error que será capaz de transmitir el canal; viene dado por:

$$C = W \log_2 \left(1 + \frac{S}{N} \right) = W \log_2 \left(1 + \frac{E_b}{N_0} \frac{R_I}{W} \right)$$

$C = R_I$ limite de la capacidad del canal, todos los bits/s están libres de error

$$\frac{C}{W} = \log_2 \left(1 + \frac{E_b C}{N_0 W} \right) \rightarrow 1 = \frac{E_b}{N_0} \cdot \log_2 \left(\left(1 + \frac{E_b C}{N_0 W} \right)^{\frac{1}{\frac{E_b C}{N_0 W}}} \right) \xrightarrow{x = \frac{E_b C}{N_0 W}} 1 = \frac{E_b}{N_0} \cdot \log_2 (1+x)^{1/x}$$

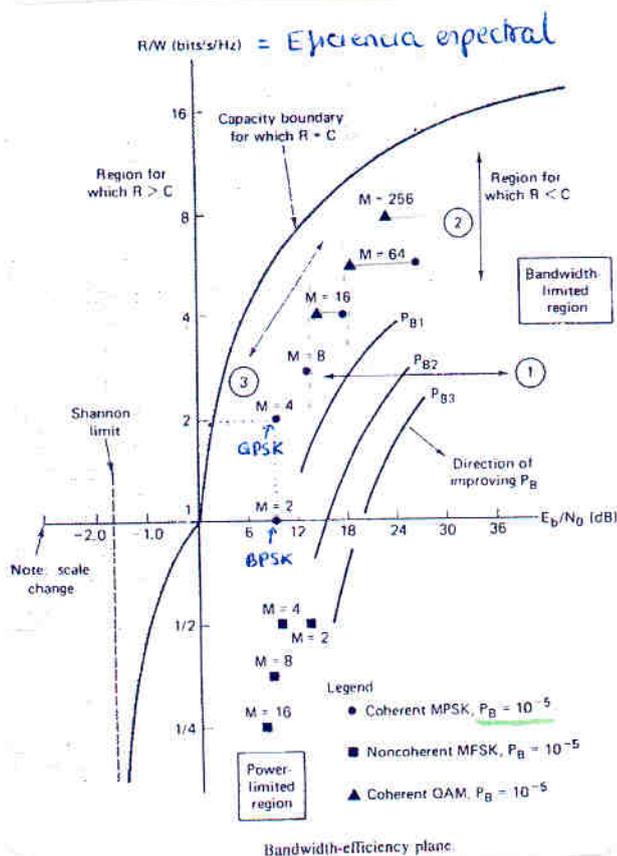
$$\lim_{x \rightarrow 0} (1+x)^{1/x} = e \rightarrow \lim_{x \rightarrow 0} \frac{E_b}{N_0} \cdot \log_2 (1+x)^{1/x} = \frac{E_b}{N_0} \log_2 e$$

Cuando $C \rightarrow 0 \Rightarrow 1 = \frac{E_b}{N_0} \log_2 e \rightarrow \frac{E_b}{N_0} = \frac{1}{\log_2 e} = 0.693 = -1.59 \text{ dB}$

Canal sin capacidad para transmitir sin error

En el caso mejor $C = R_I$ $\frac{E_b}{N_0} = -1.59 \text{ dB}$ limite de Shannon

2.3.2. Plano de eficiencia de ancho de banda



Dibujando la función

$$C = W \log_2 \left(1 + \frac{E_b}{N_0} \frac{R_I}{W} \right)$$

Sobre un plano en el cual

eje x : E_b/N_0

eje y : R_I/W = eficiencia espectral

Tendremos que para $R = C$ existe un limite que no podemos superar

Todas las modulaciones vistas hasta ahora, tomando una cierta P_b , pueden representarse como un punto en este plano.

• En el diagrama se ven algunos tipos de modulación, todas tomando $P_b = 10^{-5}$
 Si tomáramos una mejor P_b (más baja) los puntos se desplazarían hacia la dirección marcada como "Direction of improving P_b ", es decir, hacia E_b/N_0 necesario mayor.

• El eje x es E_b/N_0 . Cuanto más hacia la derecha significa que la modulación requiere mayor (S/N) en el receptor, y por tanto mayor potencia transmitida \Rightarrow Peor.

• El eje y es $R/w =$ Eficiencia espectral $[\frac{\text{bits/s}}{\text{Hz}}]$
 cuanto más hacia arriba, mayor eficiencia espectral \Rightarrow mejor

• PSK

- BPSK y QPSK requieren misma potencia
 BPSK tiene eficiencia espectral 1, QPSK tiene el doble
 \Rightarrow Es mejor QPSK

- A partir de QPSK, aumentar M implica aumentar la eficiencia espectral a costa de requerir más potencia para mantener la P_b . (requiere mayor (S/N) en receptor)

• QAM vs PSK \nearrow misma eficiencia pero MQAM menor $\frac{E_b}{N_0}$ que MPSK ya que reparte mejor su constelación.

- Se ve que los QAM están a la izquierda que los PSK \rightarrow MQAM tiene la misma eficiencia espectral, pero requiere menos potencia que PSK.

- El comportamiento de QAM al aumentar M es el mismo que PSK, i.e. eficiencia a costa de potencia requerida.

• FSK

- Al aumentar el número de niveles M en MFSK, tenemos mayor número de tonos y por tanto (aunque una M mayor implique un T_s mayor y por tanto haga que los tonos estén más cerca, no compensa) requieren ancho de banda mayor.

- Al aumentar M, para mantener P_b vemos que la $\frac{E_b}{N_0}$ requerida disminuye
 i.e. disminuir potencia a costa de disminuir eficiencia espectral

Por tanto:

PSK, QAM \Rightarrow M \uparrow \rightarrow Eficiencia espectral a costa de potencia
 Utilizaremos estos cuando el ancho de banda sea el factor limitante.

FSK \Rightarrow M \uparrow \rightarrow menor (S/N) requerida a costa de mayor ancho de banda.
 Utilizaremos estos cuando la potencia sea limitante.

ej: entre naves espaciales, dadas las enormes distancias, la potencia recibida es poquísima, y sin embargo sobra ancho de banda (hay pocas naves espaciales)

\Rightarrow usar cosas tipo 256-FSK

Significado de la curva capacity boundary

En nuestro caso, canal AWGN
(no es canal binario simétrico)
se tiene

si $R_I \leq C \Rightarrow P_e$ tan pequeña como
queramos

siendo

$$C = W \cdot \log_2 \left(1 + \frac{S}{N} \right)$$

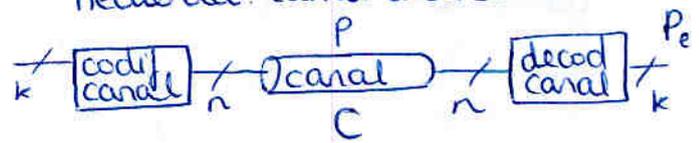
Haciendo $R_I = C$ (tope máximo de
velocidad) y sabiendo $\frac{S}{N} = \frac{E_b}{N_0} \cdot \frac{R_I}{W}$

se obtiene

$$\frac{E_b}{N_0} = \frac{2^{\frac{R_I}{W}} - 1}{R_I/W}$$

que es la curva que aparece

Recuerda: canal discreto:



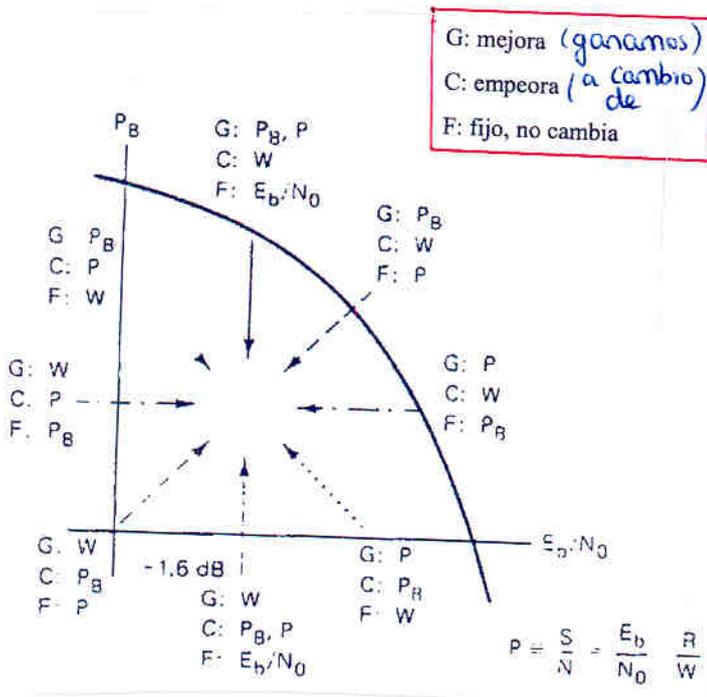
Raño $\frac{k}{n}$
capacidad del canal C

si $\frac{k}{n} \leq C \Rightarrow P_e$ (prob error residual)
tan pequeña como
queramos

Desplazamiento por las gráficas

(Siempre mantenemos constante R_I : régimen binario)

• Gráfica de P_b vs E_b/N_0



Leyenda:

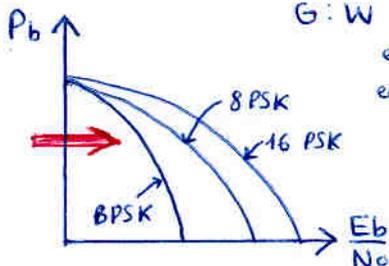
P_b : prob. error de bit

P : potencia necesaria. La interpretamos como $(\frac{S}{N})$ necesaria

W : ancho de banda

Las conclusiones son validas para todas las modulaciones

ejemplo:



F: P_b

• mantenemos cte P_b

G: W

• Ganamos en W

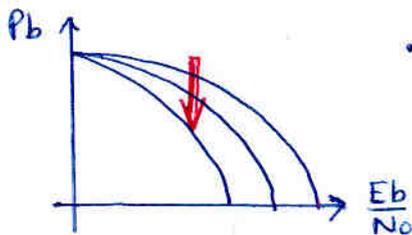
en PSK : más niveles \rightarrow mayor $T_s \rightarrow W = \frac{1+\alpha}{T}$ menor

en FSK : menos niveles \rightarrow menor W

C: P

• a cambio de consumir más potencia (dejar fija P_b e incrementar niveles implica mayor potencia)

ejemplo:



• Deja fija E_b/N_0

• Logro mejorar P_b y P (potencia = relación señal a ruido necesaria)

dejando R_I fijo!!

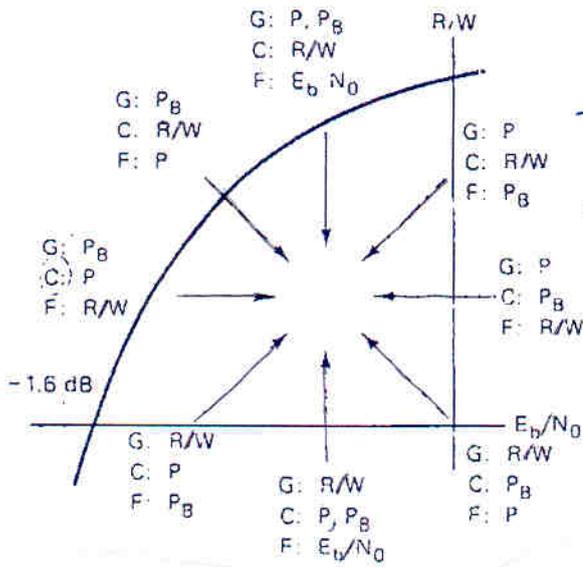
• A costa de W

en PSK \rightarrow menos símbolos \rightarrow mayor W

en FSK \rightarrow más símbolos \rightarrow mayor W

Deberías ser capaz de razonar cada dirección de desplazamiento.

• Plano de eficiencia ancho de banda



- P_b fija: recuerda curvas de P_b de
- Gano en potencia: reduce E_b/N_0
- A costa de R/W : menor eficiencia
i.e. si reducimos E_b/N_0 pero hay que mantener P_b , habrá que disminuir nº de niveles



Tema 5. Canales Preguntas de Test

1. Sistema BPSK

- $R_I = 10 \text{ Mbps}$
- $S_0(t) = A \cos \omega_0 t$ $A = 0.05 \text{ V}$
- $S_1(t) = -A \cos \omega_0 t$ $N_0 = 10^{-11} \text{ W/Hz}$
- Detección coherente con filtro adaptado
- Señal normalizada a $1 \sqrt{2}$ } (lo que llevamos haciendo toda la asignatura nosotros hacemos)
 $P = \frac{V^2}{R}$ $P = V^2$

Saber de memoria

• deducir rápidamente:

$$P_{b \text{ BPSK}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$P_b = Q\left(\frac{d/2}{\sigma}\right) = \left\{ \begin{array}{l} d/2 = \sqrt{E_b} \\ \sigma = \sqrt{N_0/2} \end{array} \right\} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

$$\left\{ \begin{array}{l} E_0 = A^2 \int_0^T \cos^2 \omega_0 t \, dt = A^2 \int_0^T \left(\frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t) \right) dt = \frac{A^2 T}{2} \\ E_1 = E_0 \end{array} \right.$$

se anulará ya que son ortogonales

$$E_b = \frac{A^2 T}{2} \Rightarrow P_b = Q\left(\sqrt{\frac{A^2 T}{N_0}}\right)$$

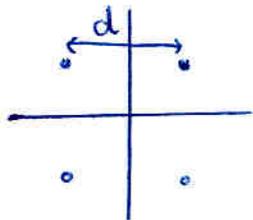
utilizando la aproximación

$$Q(x) \approx \frac{1}{\sqrt{2\pi} x} e^{-\frac{x^2}{2}}$$

2. Sistema QPSK

- Detección coherente
- Distancia entre simb más próximos = d
- Desviación típico de AWGN = σ

¿cual es la probabilidad de decidir en recepción el símbolo del cuadrante opuesto?



$$P(n_x < \frac{d}{2}, n_y < \frac{d}{2}) = P(n_x < \frac{d}{2}) \cdot P(n_y < \frac{d}{2})$$

$$= Q\left(\frac{d}{2\sigma}\right) \cdot Q\left(\frac{d}{2\sigma}\right)$$

$$P = Q^2\left(\frac{d}{2\sigma}\right)$$

3. Sistema MFSK

- Coherente
- Probabilidad de error de símbolo P_s

$$\frac{E_b}{N_0} = 4 \quad M = 8$$

$$P_s \approx (M-1) \cdot Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

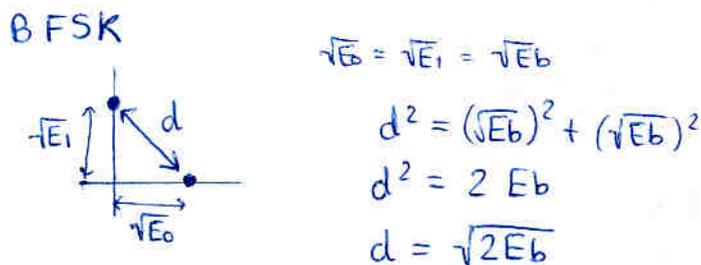
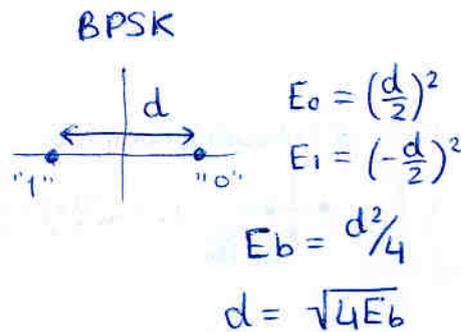
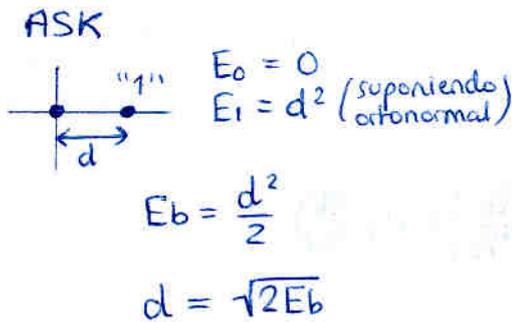
Calcular P_b

Calculamos P_s y luego

$$K = \log_2 M \rightarrow P_b = P_e \cdot \frac{2^{k-1}}{2^k - 1} = P_e \cdot \frac{2^k \cdot 2^{-1}}{2^k - 1} = P_e \frac{M/2}{M-1}$$

4. Para misma E_b ¿qué modulación binaria obtiene menor d ?

- a) PSK
- b) ASK
- c) FSK
- d) ASK y FSK



conclusión:
la ASK y la BFSK tendrán la misma probabilidad de error de bit

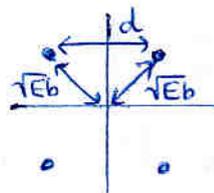
5. ¿Qué ventajas se obtiene utilizando codificación Gray en una MFSK?
Ninguna ventaja. En MFSK todos los puntos son vecinos de todos

6. QPSK con $d = 1 \text{ Julio}^{1/2}$
d.e.p. AWGN = $\frac{N_0}{2} = 0.015625 \text{ W/Hz}$

¿Cuál es P_b ?

$$P_{b \text{ QPSK}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

no queda más que sustituir



$$(\sqrt{E_b})^2 = \left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\right)^2$$

$$E_b = \frac{d^2}{2} = \frac{1}{2}$$

7. M PSK, detección coherente, ruido AWGN

- a) Gray: $P_b \approx P_e$
- b) Natural: $P_b \approx P_e$
- c) Gray $P_b \approx \frac{1}{\log_2 M} P_e$ ← correcta
- d) Natural $P_b \approx \frac{1}{\log_2 M} P_e$

8. Desvanecimiento de Rayleigh es característico de canales:
Multicamino sin propagación / visión directa.

Tema 6. Sincronización

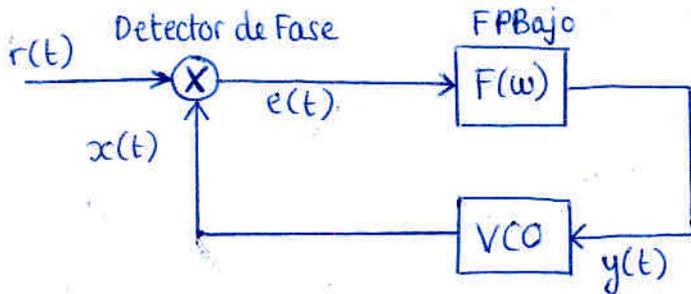
1. Introducción

Posibles soluciones:

- Tx paralelamente a la señal una señal de sincronismo: requiere mas potencia y mas ancho de banda.
- Extraer el sincronismo de la propia señal de información: es lo que haremos.

2. Sincronismo de fase

2.1 El PLL



Un posible modelo es este 'extraño' sistema, en el que la entrada y la salida son fases de $r(t)$ y $x(t)$

si $r(t)$ = una senoide

$$r(t) = \cos(\omega_c t + \theta(t))$$

la señal $x(t)$ "copia" a $r(t)$ haciendo

$$x(t) = -2 \sin(\omega_c t + \phi(t))$$

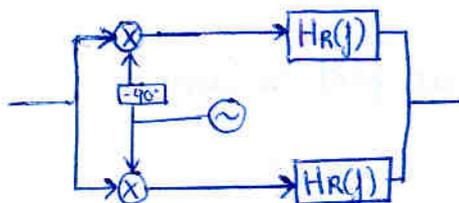
El detector de fase pretende obtener el error de fase

$$e(t) \approx \theta(t) - \phi(t)$$

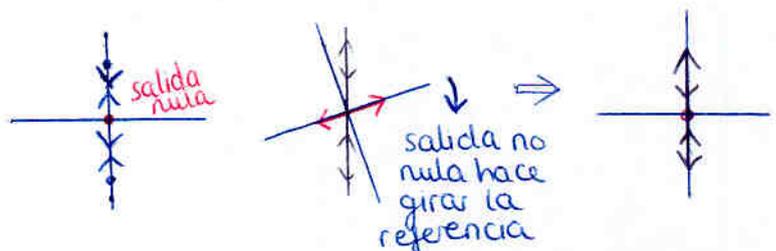
Se dice que el PLL está enganchado ($x(t)$ ha logrado "copiar" a $r(t)$) cuando $\theta(t) = \phi(t)$.

observa que en ese estado enganchado, $r(t)$ y $x(t)$ están en realidad desfasados 90° (uno es coseno y otro es seno)

Símil: es como una rama de demodulador IQ que proyecta lo que le llega sobre el eje de componente en cuadratura.



Pero en el caso del PLL, la realimentación hace que cuando esa componente no sea nula se gire el 'eje' de referencia tal que se elimine dicha componente en cuadratura.



Estudiándolo matricamente

- Suponiendo entrada senoidal:

$$r(t) = \cos(\omega_c t + \phi(t)) \quad f_R(t) = \frac{1}{2\pi} (\phi'(t) + \omega_c)$$

- consideramos por hipótesis

$$x(t) = 2 \sin(\omega_c t + \theta(t)) \quad \theta(t) = K_o \cdot y(t) \cdot t$$

- Tras el mezclador:

$$e(t) = r(t) \cdot x(t) \\ = \underbrace{\sin(\phi(t) - \theta(t)) - \sin(2\omega_c t + \theta(t) + \phi(t))}_{\text{alta frecuencia, suma de las de entrada}}$$

$$\theta'(t) = K_o \cdot y(t) \quad \leftarrow \text{efecto del VCO}$$

- Filtrando paso bajo

$$y(t) = \sin(\phi(t) - \theta(t)) * f(t)$$

que, si $\phi(t) - \theta(t)$ es pequeño

$$y(t) \approx [\phi(t) - \theta(t)] * f(t)$$

- Aplicando transformada de Fourier

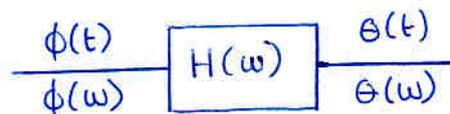
$$y(t) = [\phi(t) - \theta(t)] * f(t)$$

$$\theta'(t) = K_o \cdot y(t) \quad \leftarrow \text{por def. del VCO}$$

$$\downarrow \\ Y(\omega) = [\phi(\omega) - \theta(\omega)] \cdot F(\omega)$$

$$\downarrow \\ j\omega \theta(\omega) = K_o \cdot Y(\omega)$$

$$H(\omega) = \frac{\theta(\omega)}{\phi(\omega)} = \frac{K_o F(\omega)}{j\omega + K_o F(\omega)}$$



Corresponde a la ecuación de nuestro curioso sistema donde la entrada y la salida son fases.

Es la expresión de un filtro paso bajo
i.e. si ω (velocidad a la que varía la fase con el tiempo) no es demasiado alta, $H(\omega) = 1$ y en ese caso

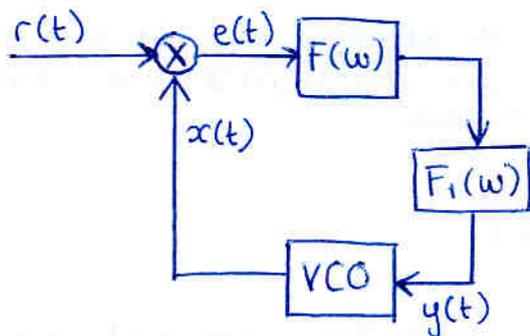
$$\theta(t) = \phi(t)$$

↓

se logra exactamente lo que pretendíamos

2.2 Respuesta del PLL ante variaciones a la entrada

Haremos uso de un nuevo filtro $F_1(\omega)$ en serie con el anterior



siendo $F_1(\omega)$ n integradores

$$F_1(\omega) = \left(\frac{1}{j\omega}\right)^n$$

$$H(\omega) = \frac{K_0 \cdot F(\omega)}{j\omega^{n+1} + K_0 F(\omega)}$$

Y vamos a estudiar si se cumple $\lim_{t \rightarrow \infty} [\phi(t) - \theta(t)] = 0$ cuando ocurren diversos cambios a la entrada.

Estudiar este límite en el tiempo es complejo, así que usaremos el teorema del valor final.

$$\lim_{t \rightarrow +\infty} g(t) = \lim_{j\omega \rightarrow 0} j\omega G(\omega)$$

$$\lim_{j\omega \rightarrow 0} j\omega [\phi(\omega) - \theta(\omega)] = \lim_{j\omega \rightarrow 0} j\omega [\phi(\omega) - H(\omega)\phi(\omega)] = \lim_{j\omega \rightarrow 0} j\omega [1 - H(\omega)]\phi(\omega)$$

$$= \lim_{j\omega \rightarrow 0} \underbrace{\frac{j\omega^{n+2}}{j\omega^{n+1} + K_0 F(\omega)}}_{A(\omega)} \cdot \phi(\omega) = \lim_{j\omega \rightarrow 0} A(\omega) = 0?$$

Si estudiamos este límite en los siguientes casos:

Salto de fase : $\phi(t) = \Delta\phi_0 \cdot u(t) \xrightarrow{\text{se obtiene}} \lim_{j\omega \rightarrow 0} A(\omega) = 0$ si $n \geq 0$

Salto de frecuencia : $\phi(t) = \Delta\omega_0 \cdot t \longrightarrow \lim_{j\omega \rightarrow 0} A(\omega) = 0$ si $n \geq 1$

Rampa de frecuencia : $\phi(t) = \Delta\omega_0 \cdot t \cdot t \longrightarrow \lim_{j\omega \rightarrow 0} A(\omega) = 0$ si $n \geq 2$

Intuitivamente:

- Ante un cambio de fase, en $y(t)$ habrá una tensión positiva que hará aumentar la frec de $x(t)$ hasta igualar la fase, donde la frec de $x(t)$ volverá a bajar ya que $y(t)$ se anula.
- Ante un cambio de frecuencia en $r(t)$, para que la frecuencia aumente también en $x(t)$, el VCO necesitará a su entrada una tensión $y(t) \neq 0$
- $n=0$ \hookrightarrow Sin integrador ($n=0$) necesitaríamos error de fase $\phi(t) - \theta(t) \neq 0$ tal que $y(t)$ sea distinta de cero. i.e. sin integrador igualamos la frecuencia pero a costa de un desfase
- $n=1$ \hookrightarrow Con un integrador, suponer un condensador, éste guardará la tensión necesaria para $y(t)$ para que el VCO oscile a la nueva frec; pudiendo tener $\phi(t) - \theta(t) = 0$ siendo el integrador el q. proporciona la $y(t) \neq 0$

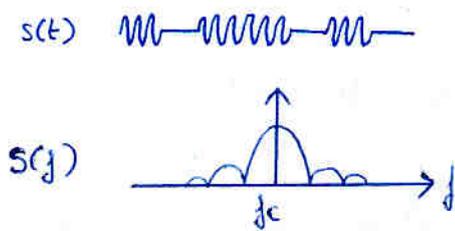
2.3 Aplicaciones del PLL

Demodulador de Frecuencia:

Viendo como copia el PLL la frecuencia de entrada (aún variando ésta) parece obvio que en $y(t)$ tendremos una señal proporcional a la frecuencia de entrada

Modulaciones con portadora residual

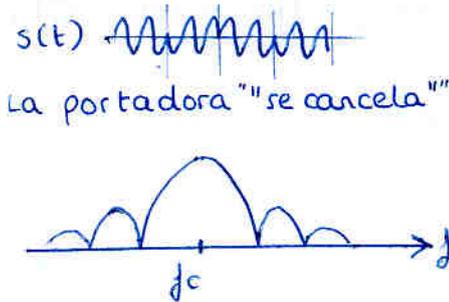
ejemplo OOK



El PLL "por naturaleza" se engancharía a la portadora f_c (por tener mucha potencia) obteniendo por lo tanto una copia de la portadora con sincronismo en fase, que es lo necesario para demodular.

Modulaciones con portadora suprimida

ejemplo BPSK



La portadora "se cancela"

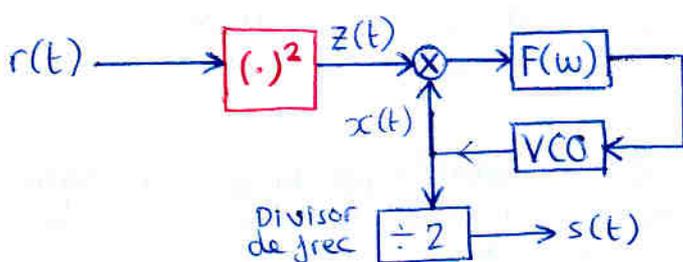
Lo único que haría el PLL visto hasta ahora sería copiar esos cambios de fase

Necesitamos un demodulador para este tipo de modulaciones.

2 formas

- Elevar al cuadrado: permite un detector de portadora que podremos utilizar para demodular
- Bucle de costas
Ya actúa como demodulador y además no eleva al cuadrado

Detector de portadora: elevar al cuadrado (bucle cuadrático)



$r(t)$ señal BPSK

$$r(t) = \cos(\omega_0 t + \frac{\pi \cdot m(t)}{\phi(t)}) \quad m(t) = 0 \text{ ó } 1$$

al elevar al cuadrado

$$z(t) = \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t + 2\pi \cdot m(t))$$

$$= \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t)$$

Esa $z(t)$ es copiada a $x(t)$ por el PLL, donde tras dividir la frecuencia entre 2 se obtiene senoide de frecuencia ω_0 .

Problema al elevar al cuadrado:

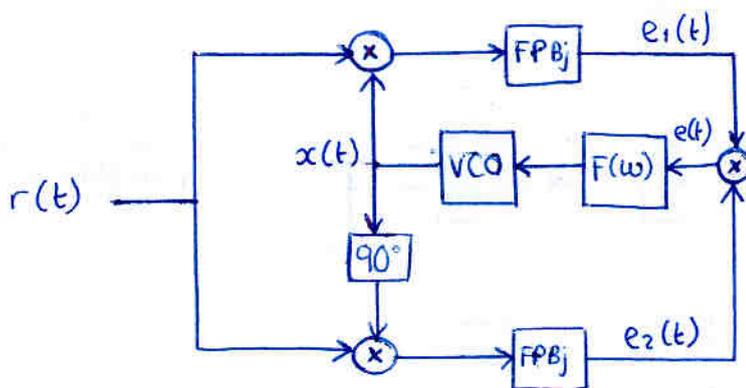
- El ruido también se eleva al cuadrado: i.e. se cuadruplica su varianza.
- La portadora obtenida tendrá variaciones de amplitud y frecuencia: **JITTER**

Situación de compromiso:

- Ancho de banda de $F(\omega)$
 - alto: el PLL se engancha muy rápido ante estímulo PERO dejamos pasar más ruido \Rightarrow Jitter.

Bucle de costas

Logra demodular modulaciones de portadora suprimida sin tener que elevar al cuadrado. (requiere menor SNR)



A la entrada, BPSK

$$r(t) = \cos[\omega_c t + \pi m(t) + \beta(t)]$$

Fase normalmente de con el tiempo, aunque en general puede variar ej: efecto Doppler.

se obtendrá $x(t) = -2 \sin(\omega_c t + \theta(t))$

Si calculamos $e_1(t)$ y $e_2(t)$ y las multiplicamos, se obtiene $e(t) = e_1(t) \cdot e_2(t) = \dots \approx \beta(t) - \theta(t)$

Y el resto se comportará como un PLL donde la señal de error de fase es $\beta(t) - \theta(t)$, es decir que $x(t)$ será

$$x(t) = -2 \sin(\omega_c t + \theta(t)) \quad \text{con } \theta(t) = \beta(t)$$

\Downarrow
Sincronismo de fase

Intuitivamente: $e_1(t)$ y $e_2(t)$ son las componentes en cuadratura y en fase. Por tanto ante una BPSK (que posee en ocasiones comp en fase y otras en cuadratura) siempre tendrá una de las dos proyecciones nulas, que al multiplicarse dar $e(t)$ nulo. La realimentación se encargará de ajustar la referencia para que esto sea así.

Además si miramos $e_2(t)$

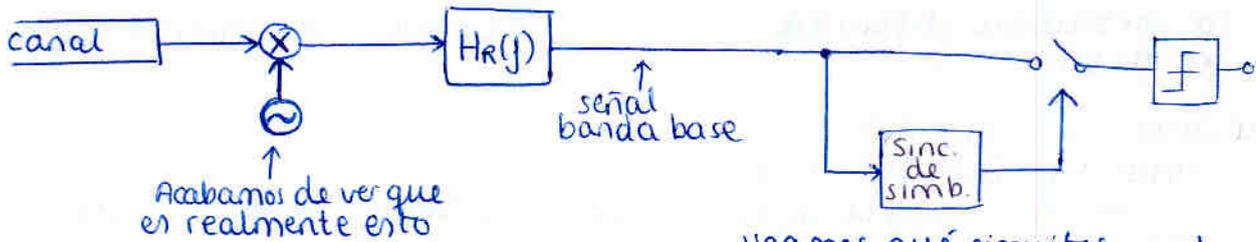
$$e_2(t) = -\cos(\pi m(t) + \beta(t) - \theta(t))$$

$$= -\cos(\pi m(t))$$

$$= \begin{cases} -1 & m(t) = 0 \\ 1 & m(t) = 1 \end{cases} \Rightarrow \text{Demodulador}$$

En efecto se comprueba que $e_2(t)$ es la componente en cuadratura i.e. demodulador BPSK.

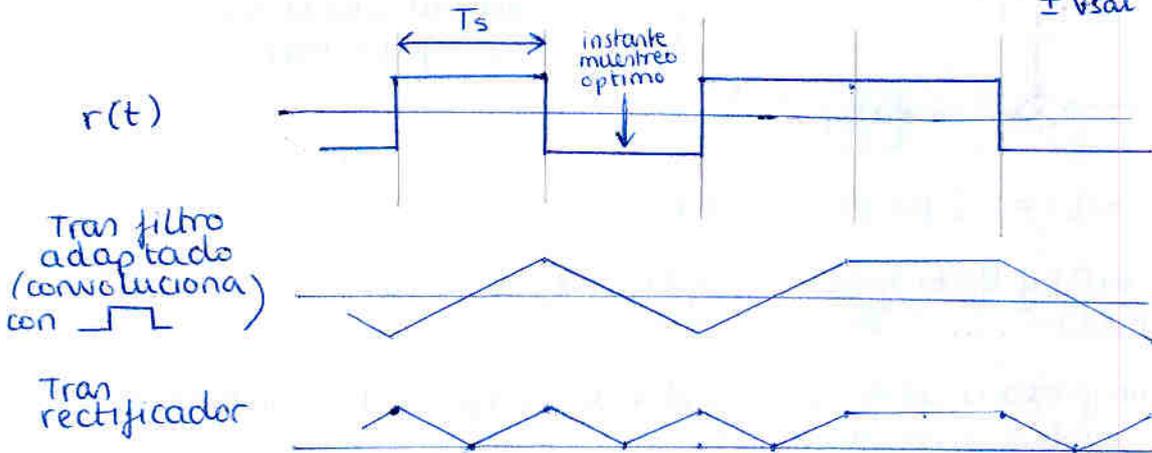
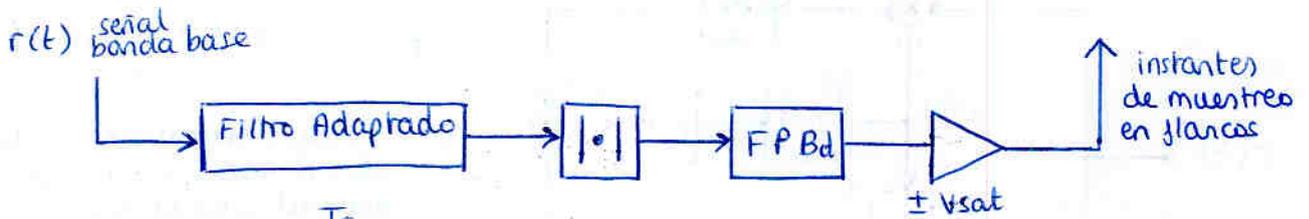
3. Sincronismo de simbolo



veamos que circuitos pueden proporcionarnos los instantes de muestreo a partir de la señal en banda base.

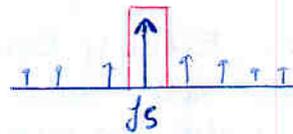
3.1 Sincronismo en lazo abierto

• Sincronizador con filtro adaptado

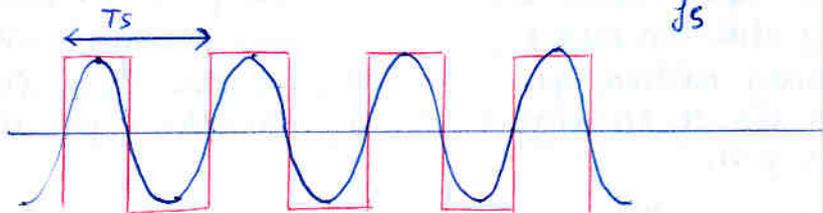


siendo $\text{[rectified signal]} = \text{[periodic signal]} + \text{[DC component]}$

con el filtro paso banda cogemos sólo la componente en f_c obteniendo un seno



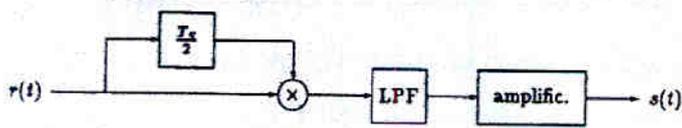
Tran F.P.Bd



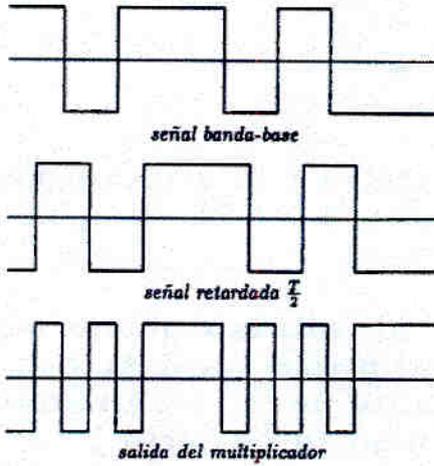
Con el amplif $\pm V_{sat}$

Podemos usar los flancos de la señal cuadrada para muestrear. Tal vez haya un desfase, pero será determinista y no habrá más que "coger un destornillador y... ajustarlo"

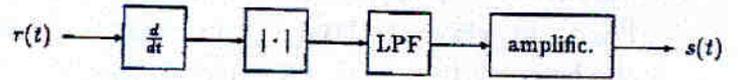
• sincronizador con retardo



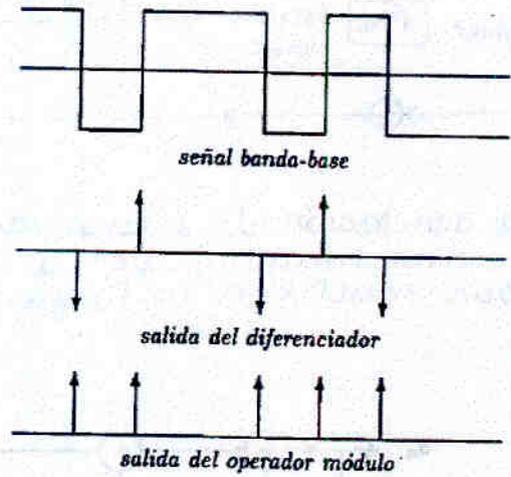
Sincronizador de símbolo con retardo.



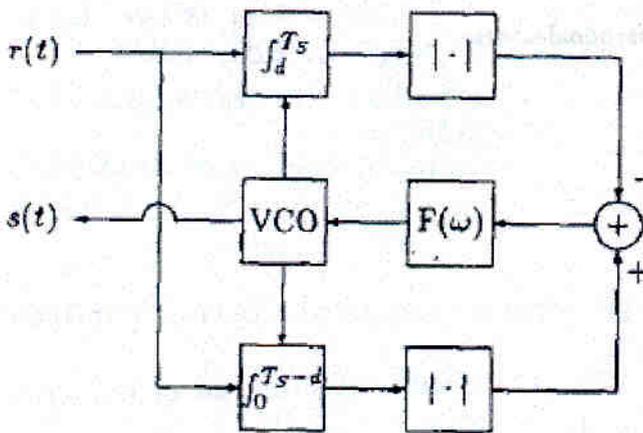
• Sincronizador con diferenciador



Sincronizador con diferenciador.



3.2. Sincronismo en lazo cerrado



Bucle con puerta early/late.

• Bucle Early-Late

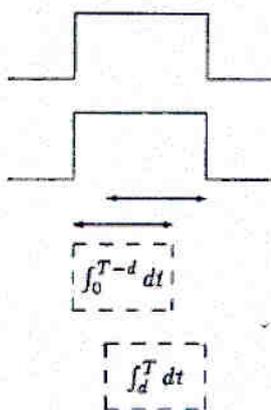
La señal que sale del VCO x r(t) se integra en 2 intervalos distintos

← y se calcula la diferencia.

El resultado será proporcional a la desincronización entre la señal de datos y la de referencia para el muestreo.

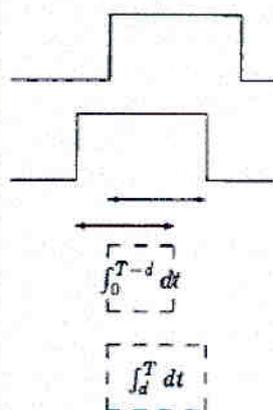
La realimentación se encargará de que este valor sea nulo, sincronizando así los datos con la referencia

sincronizado



diferencia nula

ligero desincronismo



diferencia negativa

señal de datos

señal de ref.

[d, T_s] y [0, T_s - d]

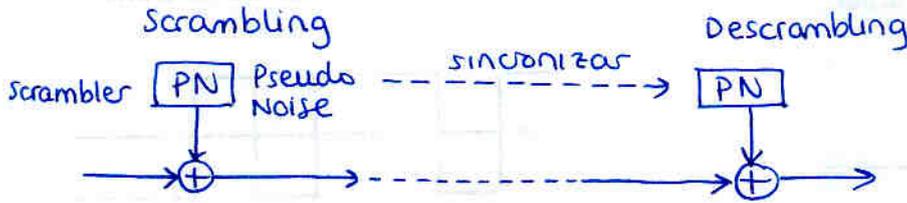
integración early

integración late

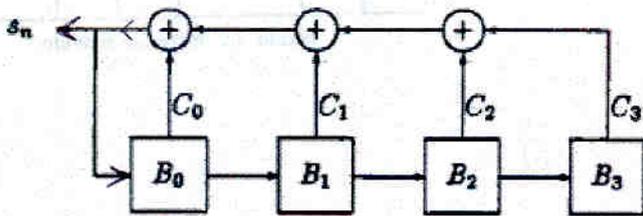
intervalos de integración

3.3 Aleatorización

Todos los esquemas anteriores requieren que la señal tenga viveza (que no sea tipo 11111...111)
 Para evitar estas posibles situaciones, se aleatorizan los datos antes de tx sumándoles señal pseudoaleatoria



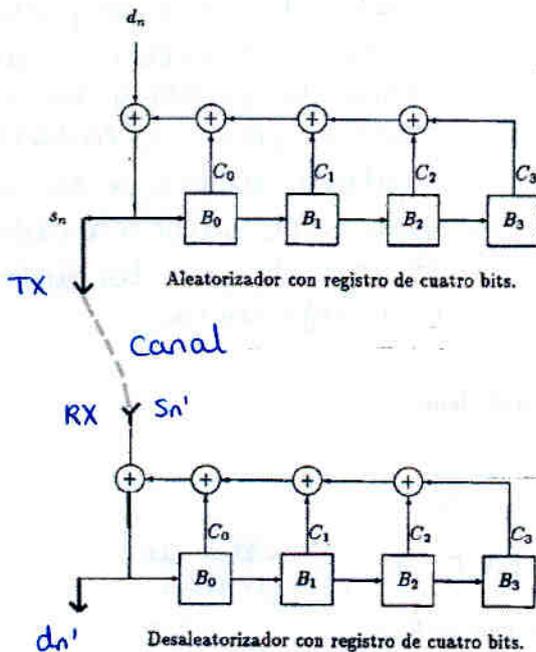
La generación de secuencias pseudoaleatorias se hace mediante circuitos muy similares a los generadores de CRC y que se usan mucho a lo largo de la carrera



Esquema de generación de una secuencia pseudoaleatoria.

- En realidad genera una secuencia periódica, pero de periodo muy largo (se busca cuanto mas largo mejor)
- No todas las combinaciones iniciales son válidas
 ej 0000 se queda SIEMPRE en 0000
 algunas pueden derivar al estado nulo.
- No todas las conexiones son válidas
 (recuerda $C_i = 0 \rightarrow$ no conexión)

- Sincronizar secuencia aleatoria



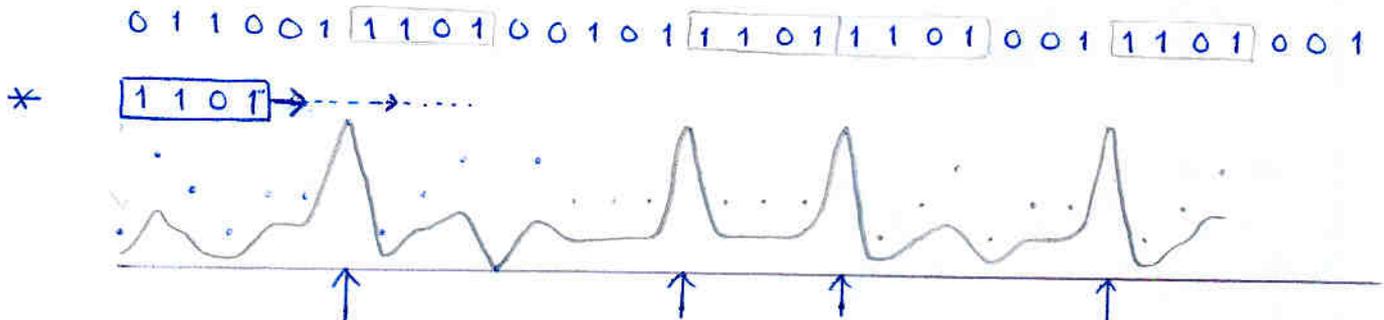
se utiliza esquema autosincronizante
 si el estado inicial es el mismo,
 $d_n' = d_n$ (siempre que no haya errores)

4. Sincronismo de trama

• No podemos delimitar la trama físicamente

• se hace mediante secuencias delimitadoras

Un método para hallar la secuencia es ir haciendo la correlación con ésta.



Hay picos en la correlación cuando la secuencia aparece, de valor igual a la autocorrelación

si la secuencia aparece en los datos, ésta se repite para diferenciarla (character stuffing)

Hay algunas secuencias de bits óptimas estadísticamente

Secuencia de Barker

1
11 0 10
110
1110 ó 1101
11101
⋮

Calculada teóricamente

Secuencias cuya autocorrelación se parezca más a una delta de Dirac

No tiene en cuenta los bits adyacentes

Bromilla para acordarse

- teórico = muy ladrador (barker)
- no tiene en cuenta bits adyacentes: poco mbrador

Secuencia de Willard

1
10
110
1100
11010
⋮

La que obtenía mejores resultados experimentales teniendo en cuenta todas las posibilidades de bits adyacentes.

1. The first part of the document is a list of names.

2. The second part is a list of dates.

3. The third part is a list of locations.

4. The fourth part is a list of events.

5. The fifth part is a list of people.

6. The sixth part is a list of organizations.

7. The seventh part is a list of activities.

8. The eighth part is a list of places.

9. The ninth part is a list of things.

10. The tenth part is a list of people.

11. The eleventh part is a list of events.

12. The twelfth part is a list of people.

13. The thirteenth part is a list of events.

14. The fourteenth part is a list of people.

15. The fifteenth part is a list of events.

16. The sixteenth part is a list of people.

17. The seventeenth part is a list of events.

18. The eighteenth part is a list of people.

19. The nineteenth part is a list of events.

20. The twentieth part is a list of people.

100

101

102

103

104

105

106

107

108

109

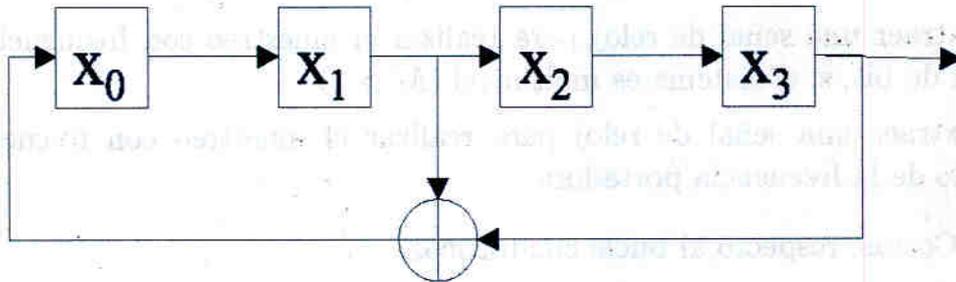
110

Tema 6 - Sincronización

Preguntas de Test

17. El subsistema de extracción de sincronismo de símbolo de un receptor...
- a. no es necesario que exista, si el receptor es no coherente.
 - b. puede ser en bucle abierto o en bucle cerrado. ✓
 - c. debe extraer una señal de reloj para realizar el muestreo con frecuencia igual a la frecuencia de bit, si el sistema es multinivel ($M > 2$).
 - d. debe extraer una señal de reloj para realizar el muestreo con frecuencia igual a un múltiplo de la frecuencia portadora.
18. El bucle de Costas, respecto al bucle cuadrático...
- a. necesita menor SNR para mantener el enganche de fase. ✓
 - b. necesita mayor SNR para mantener el enganche de fase. *ya que no eleva al cuadrado*
 - c. necesita la misma SNR para mantener el enganche de fase.
 - d. opera a una frecuencia doble.
19. La función de transferencia del PLL...
- a. interesa que tenga un ancho de banda pequeño para realizar un rápido seguimiento.
 - b. interesa que tenga un ancho de banda grande para realizar un rápido seguimiento. ✓
 - c. interesa que tenga un ancho de banda grande para obtener baja potencia de ruido.
 - d. Ninguna de las demás es cierta.
19. El bucle de Costas sirve para:
- a. obtener una señal de sincronismo de fase a partir de una señal con portadora residual
 - b. obtener una señal de sincronismo de fase a partir de una señal con portadora suprimida ✓
 - c. adquisición de sincronismo en espectro ensanchado
 - d. seguimiento de sincronismo en espectro ensanchado.

15. En la configuración que se adjunta, de registros binarios con desplazamiento, ¿cuál es el período de la secuencia que se obtiene? Suponga como valor inicial el estado $(x_0, x_1, x_2, x_3) = (1, 0, 0, 0)$



- a. Período ≤ 6 .
- b. $6 < \text{Período} \leq 10$.
- c. $10 < \text{Período} \leq 15$.
- d. $15 < \text{Período}$.
16. Al generar una secuencia pseudoaleatoria mediante un registro de desplazamiento con realimentación:
- a. la secuencia nunca puede repetirse *mentira, sería un chollo*
- b. el valor cero nunca puede aparecer en la secuencia *se podría con ciertas realimentaciones 'erróneas'*
- c. la secuencia no se repetirá si el estado inicial del registro es distinto de cero
- d. la secuencia siempre se repetirá ✓

Aprender de memoria (en Transmisión de datos)

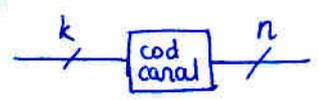
Tema 2 Teoría de la información

Huffman: Resto $\left(\frac{\text{Primer grupo}}{\text{Simbolos tras decod} - 1} \right) = \text{resto} \left(\frac{\text{simbolos de fuente}}{\text{simbolos tras decod} - 1} \right) \quad \bar{n} \geq H(X)$

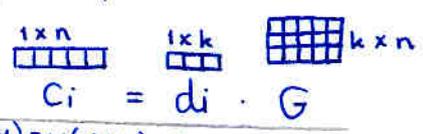
Capacidad: $C = \max_{H(X)} (I(X;Y)) \xrightarrow{BSC} C = 1 - H_2(p)$ Teorema de codificación de canal de Shannon

$H_2(P_e) \geq 1 - \frac{C}{R} \rightarrow R = \frac{k}{n}$ en general $k < n$
 $R < 1$

Tema 3 Control de errores



Códigos Bloque Lineales:



$G \rightarrow G_{\text{sistem}} = \begin{pmatrix} 1 & 0 & 0 & X & X \\ 0 & 1 & 0 & X & X \\ 0 & 0 & 1 & X & X \end{pmatrix}$

$H_T = \begin{pmatrix} X & X \\ X & X \\ X & X \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ - d_{min} : nº filas sumar para tener cero "peso mínimo de palabras código"

Síndrome $(X, X) = C \cdot H^T$

$d(x,y) = w(x+y)$
 d_{min}
 $w(x)$: peso hamming

$w(e) \leq d_{\text{min}} - 1 \Rightarrow$ detectable
 $w(e) \leq \lfloor \frac{d_{\text{min}} - 1}{2} \rfloor \Rightarrow$ corregible

Códigos cíclicos:

No sistemático: $c(x) = m(x) \cdot g(x) \rightarrow c = m \cdot G_{\text{ns}}$

$G_{\text{ns}} = \begin{pmatrix} x^{k-1} \cdot g(x) \\ x^k \cdot g(x) \\ \vdots \\ x \cdot g(x) \\ g(x) \end{pmatrix} = \begin{pmatrix} \boxed{} & 0 & 0 \\ 0 & \boxed{} & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & \boxed{} \end{pmatrix}$

Sistemático: $c(x) = x^{n-k} \cdot m(x) + r(x) \leftarrow c = m \cdot G_s$

$r(x) = \text{resto} \left(\frac{x^{n-k} \cdot m(x)}{g(x)} \right)$

$r(x)$: $n-k$ bits \Rightarrow grado $n-k-1$
 el resto tiene grado \leq divisor
 g : $n-k+1$ bits \leftarrow grado $g(x) = n-k$

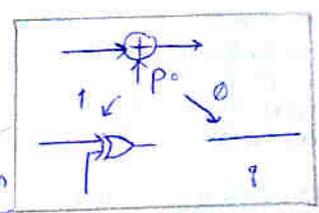
Propiedades $g(x)$: polinomio generador
 grado $n-k$
 $g(x) = x^{n-k} + \dots + 1$
 i.e. 1er y último coef $\neq 0$
 $\text{resto} \left(\frac{x^n + 1}{g(x)} \right) = 0$ $n+1$ bits (mas q el qto de palabras código)

Propiedades: DETECCIÓN (no corrección)

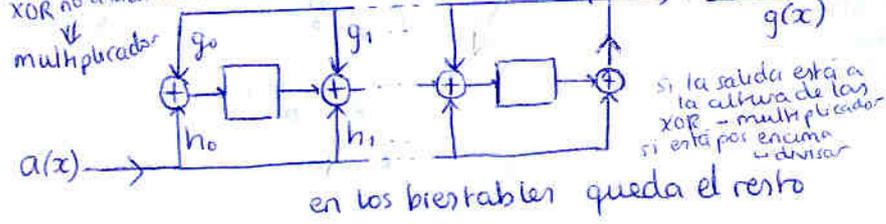
- detectan TODOS errores simples $\Rightarrow d_{\text{min}} > 1$
- detectan TODAS rajagas de long $n-k$
- prob no detectar rajaga long $n-k+1 \rightarrow P_{nd} = \frac{1}{2^{n-k+1}}$ (i.e. sólo $\frac{1}{4}$ de esas 2^{n-k+1} no son detectables)
- prob no detectar rajaga long $n-k+2 \rightarrow P_{nd} = \frac{1}{2^{n-k}}$

$\text{resto} \left(\frac{c(x)}{g(x)} \right) = 0$

Variantes: - códigos recortados
 - paridad invertida



T RUCO: XOR a la altura de FF \rightarrow coef de izda a dcha (al revés de la altura de la escitura normal de un polinomio)
 XOR no a la altura de FF \rightarrow coef tal y como se escriben.



- si $g(x)$ tiene como factor $x^h + 1 \Rightarrow$ se detectan todos los errores impares
- si $g(x)$ tiene como factor un polinomio primitivo \Rightarrow se detectan todos los errores dobles

Códigos convolucionales

$$C(n, k, K)$$

↑ el mensaje actual y los K-1 anteriores

- cada estado se nombra según el contenido de los K-1 flip flops
- la etiqueta del arco indica la palabra código saliente

cada instante en Viterbi hay tantas ramas como n° de estados

$$2^{(K-1)k}$$

↑
n° registros

↑
bit por registro

Tema 4 - modulaciones Digitales

$H(f)$: respuesta total (filtro tx + canal + filtro rx)
 $H_{eq}(f)$: canal equivalente (repetir $H(f)$ cada $1/2T$)

· Transformada generalizada

$$\begin{matrix} \sin \\ \cos(-\theta) \\ \cos \end{matrix}$$

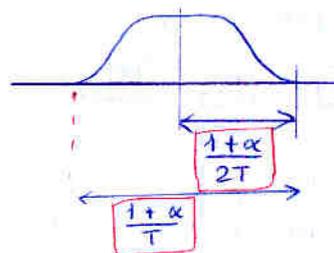
entre frecuencias:

$$\uparrow \uparrow$$

$1/T$ ó $1/2T$ si sincrónico

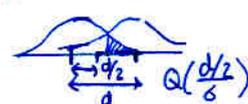
- BPSK
- QPSK
- 4GAM
- M-GAM
- FSK
- MSK
- GMSK

Pulsos Nyquist
o coseno alzado



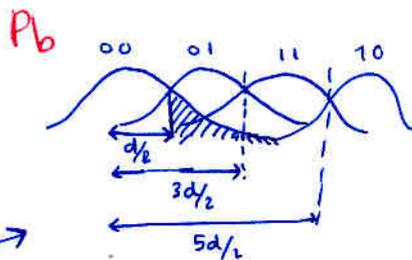
Tema 5. canales

P_e



$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx$$

← Cálculo de P_e



$$P_b | m_0 = \frac{1}{2} P(m_1 | m_0) + \frac{2}{2} P(m_2 | m_0) + \frac{1}{2} P(m_3 | m_0)$$

$$= \frac{1}{2} Q\left(\frac{d/2}{\sigma}\right) + \frac{2}{2} Q\left(\frac{3d/2}{\sigma}\right) + \frac{1}{2} Q\left(\frac{5d/2}{\sigma}\right)$$

$$P_b = P(m_0) \cdot P_b | m_0 + P(m_1) \cdot P_b | m_1 + \dots$$

$$P_b \approx \frac{P_e}{\log_2 M}$$

← código Gray
 ← suponer error sólo con símbolo contiguo

Nota en MSK

$$P_b = \frac{M/2}{M-1} P_e = \frac{2^{k-1}}{2^k - 1} P_e$$

$k = \text{n° bits}$
 $M = 2^k \text{ n° niveles}$

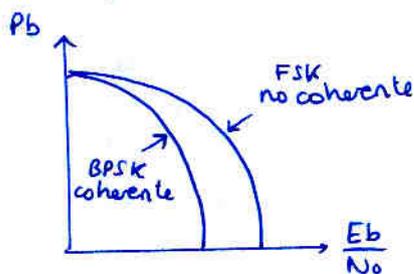
Arthur y Dim

$$Q\left(\frac{\bar{d}}{2\sigma}\right) \leq P_e \leq (M-1) Q\left(\frac{d^*}{2\sigma}\right)$$

\bar{d} : distancia media de símbolos i-ésimo con su vecino más próximo

d^* : distancia mínima entre 2 símbolos

Gráficas de P_b



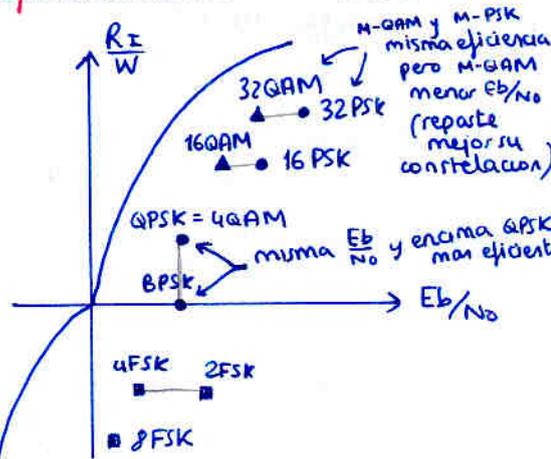
$$\left(\frac{C}{N}\right) = \frac{E_b}{N_0} \cdot \frac{R_f}{W}$$

M-FSK: al aumentar M baja P_b (a cambio → gran ancho de banda)

M-PSK y M-QAM: al aumentar M aumenta P_b (a cambio → mayor eficiencia espectral)

Plano eficiencia ancho banda

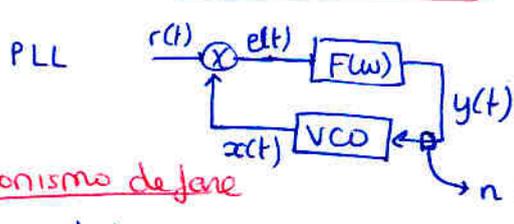
para P_b fija:



$$C = W \log_2 \left(1 + \frac{C}{N}\right)$$

cuando $R_f = C$

Tema 6. Sincronización



$$r(t) = \cos(\omega t + \theta(t))$$

$$x(t) = -2 \sin(\omega t + \phi(t))$$

$$e(t) = \theta(t) - \phi(t) \rightarrow 0$$

sincronismo de fase

Sincronismo de fase

En modulaciones con portadora o portadora residual, el PLL se engancha a la portadora por inercia.

- $x(t)$ se engancha ante un:
- $n \geq 0 \rightarrow$ salto de fase
 - $n \geq 1 \rightarrow$ salto de frecuencia
 - $n \geq 2 \rightarrow$ rampa de frecuencia

En modulaciones con portadora suprimida (ej MPSK) se necesita algo para obtener señal de sincronismo de fase.

→ Bucle cuadrático \Rightarrow al elevar al cuadrado y dividir frec por 2 se obtiene la portadora
 Problema: ruido se multiplica al cuadrado \rightarrow JITTER

$$r(t) = \cos(\omega_0 t + \pi m(t))$$

$m(t) = 0 \text{ ó } 1$

$$z(t) = \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t + 2\pi \cdot m(t))$$

$$= \frac{1}{2} + \frac{1}{2} \cos(2\omega_0 t)$$

→ Bucle de costas \rightarrow no eleva al cuadrado \rightarrow requiere menor (S/N)

2 ramas obtienen cada una una componente de la señal y mediante realimentación se logra que el producto de ambas se anule (se 'gira' la referencia para anular las componentes)



Sincronismo de símbolo

se realiza SOBRE LA SEÑAL BANDA BASE

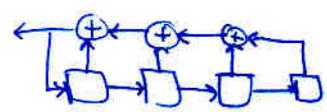
- Lazo abierto:
- Filtro adaptado + rectificador + filtro Paso Bd a f_s
 - con retardo multiplicando a la no retardada
 - con diferenciador (derivar + rectificar + filtrar componente a f_s)

- Lazo cerrado:
- Bucle early-late
 la señal de salida se multiplica por la $r(t)$ entrada y el resultado se integra en 2 intervalos (early, late)
 la diferencia de ambas integrales ajusta la salida



necesitan viveza en la señal \rightarrow aleatorización

scrambler } generador secuencias pseudoaleatorias
 descrambler } (siempre se repiten. interesa periodo largo)
 no vale el estado 0 0 0... 0
 ya que se queda muerto



Sincronismo de trama:

Ir haciendo la correlación de las secuencias delimitadoras
 secuencia Barker (teórica, no tiene en cuenta bits adyacentes)
 secuencia Willard (experimental) } secuencias óptimas estadísticamente